

# **Entwurf von Künstlichen Neuronalen Netzen zur Regelung von Prozessgrößen in einer Schmelzwanne für Flachglas**

Von der Fakultät für Maschinenbau, Elektro- und Energiesysteme  
der Brandenburgischen Technischen Universität Cottbus-Senftenberg  
zur Erlangung des akademischen Grades eines  
Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte

## **Dissertation**

vorgelegt von

Dipl.-Ing.

**Martin Handreg**

geboren am 19.02.1985 in Cottbus

Vorsitzender:	Prof. Dr.-Ing. Harald Schwarz
Gutachter:	Prof. Dr.-Ing. Gerhard Lappus
Gutachter:	Prof. Dr.-Ing. Ulrich Berger

Tag der mündlichen Prüfung: 12.07.2016



## Kurzfassung

In der vorliegenden Arbeit wird die Eignung von Künstlichen Neuronalen Netzen hinsichtlich der Modellierung des komplexen Prozessverhaltens in einer Flachglasschmelzanlage analysiert. Die Identifikation und das Training der neuronalen Prozessmodelle erfolgen mit Messdaten einer Schmelzwanne für Flachglas. Im Vordergrund steht die Evaluation einer geeigneten Netzstruktur und die Parametrierung der Netzparameter. Dabei wird der Einfluss der einzelnen Netzparameter in Bezug auf die Genauigkeit der Netze eingehend untersucht. Anhand von Testdaten wird nachgewiesen, dass die qualitätsbestimmenden Temperaturen und der Glasstand mit Künstlichen Neuronalen Netzen hinreichend genau berechnet werden können. Auf Basis der entwickelten neuronalen Prozessmodelle wird anschließend eine modellbasierte prädiktive Regelstrategie beschrieben. Neben der Auswahl des Gütekriteriums und des Optimierungsalgorithmus zur Berechnung zukünftiger Stellgrößen werden Richtlinien zur Dimensionierung der verfügbaren Reglerparameter abgeleitet.

## Abstract

This thesis analyzes the suitability of Artificial Neural Networks for modeling the complex process in a float glass melting plant. The identification and training of the neural process models are being carried out with real float glass furnace data. The focus is on the evaluation of an appropriate network structure and parameter configuration. With respect to the network precision the effect of parameter changes will be investigated and it will be shown that it is possible to calculate the relevant temperatures and the glass level in an accurate way. The results will be validated by test data. In the second step, a model-based predictive control strategy to determine future control signals is described and a selection of the performance criterion, the numerical optimization program and guidance for the available control parameters are given.



# Inhaltsverzeichnis

<b>KURZFASSUNG.....</b>	<b>I</b>
<b>ABBILDUNGSVERZEICHNIS.....</b>	<b>VI</b>
<b>TABELLENVERZEICHNIS .....</b>	<b>IX</b>
<b>VERZEICHNIS DER SYMBOLE UND ABKÜRZUNGEN .....</b>	<b>X</b>
<b>1 PROBLEMSTELLUNG UND ZIELSETZUNG.....</b>	<b>1</b>
1.1 Eigenschaften und Herstellung von Glas .....	1
1.2 Stand der Technik und Forschung .....	6
1.3 Regelungstechnische Problemstellungen .....	9
1.4 Zielsetzung und Vorgehensweise .....	11
1.4.1 Ableitung der Zielsetzung .....	11
1.4.2 Struktur der Arbeit .....	13
<b>2 ENTWICKLUNGSUMGEBUNG.....</b>	<b>15</b>
2.1 Realisierung Künstlicher Neuronaler Netze mit <i>MATLAB</i> .....	16
2.1.1 Funktionsumfang und Anwendungsgebiete .....	16
2.1.2 Grundlegendes Design.....	17
2.1.3 Konfigurationsmöglichkeiten.....	20
2.2 Datensatzanalyse .....	22
2.2.1 Struktur des Datensatzes .....	22
2.2.2 Interpretation der Messdaten.....	24
2.2.3 Aufbereitung des Datensatzes.....	33
2.3 Konfigurationsassistent und Bewertungskriterien.....	37
<b>3 IDENTIFIKATION GEEIGNETER NEURONALER PROZESSMODELLE .....</b>	<b>40</b>
3.1 Methodik.....	40
3.2 Simulationsergebnisse.....	41
3.2.1 Auswahl der Ein- und Ausgangsgrößen .....	41
3.2.2 Einfluss der Trainingsmethoden .....	45
3.2.3 Einfluss der Aktivierungsfunktionen .....	48

3.2.4	Einfluss der verborgenen Neuronen .....	52
3.2.5	Einfluss der Tapped Delay Lines .....	55
3.2.6	Einfluss der Trainings- und Testzeiträume.....	59
3.2.7	Reproduzierbarkeit der Simulationsergebnisse .....	61
3.2.7.1	Analyse der Reproduzierbarkeit bei zufälliger Initialisierung .....	61
3.2.7.2	Modifizierung der Trainingsumgebung.....	64
3.2.7.3	Verifikation der bisherigen Simulationsergebnisse.....	64
3.2.8	Vergleich der Simulationsergebnisse mit dem vereinfachten Datensatz .....	66
3.3	Zusammenfassung .....	68
<b>4</b>	<b>MULTI STEP PREDICTION.....</b>	<b>70</b>
4.1	Simulationsumgebung und Methodik .....	70
4.2	Simulationsergebnisse.....	71
4.2.1	Einfluss der verborgenen Neuronen .....	71
4.2.2	Einfluss der Tapped Delay Lines .....	72
4.2.3	Einfluss des Trainings- und Prädiktionszeitraums.....	75
4.3	Zusammenfassung .....	77
<b>5</b>	<b>REGELUNGSTECHNISCHES KONZEPT.....</b>	<b>80</b>
5.1	Ableitung einer geeigneten Regelstrategie .....	80
5.2	Neuronale prädiktive Regelung.....	85
5.2.1	Grundprinzip.....	85
5.2.2	Mathematische Beschreibung des Prädiktors.....	87
5.2.2.1	One Step Prediction .....	87
5.2.2.2	Multi Step Prediction.....	92
5.2.3	Gütekriterium.....	94
5.2.4	Optimierung nach dem Levenberg-Marquardt-Algorithmus .....	96
5.2.4.1	Entscheidungsgrundlage .....	96
5.2.4.2	Herleitung des Regelgesetzes nach Levenberg-Marquardt .....	97
5.2.4.3	Cholesky-Zerlegung .....	102
5.2.4.4	Zusammenfassung und Ablaufplan .....	103
5.2.5	Wahl der Reglerparameter .....	105
5.3	Diskussion .....	109
<b>6</b>	<b>ZUSAMMENFASSUNG UND AUSBLICK.....</b>	<b>112</b>
<b>ANHANG</b>	<b>.....</b>	<b>115</b>

A	Trainingsmethoden in <i>MATLAB</i> .....	115
B	Aktivierungsfunktionen in <i>MATLAB</i> .....	117
C	Aufbereitung der verfügbaren Messgrößen.....	120
D	Grafische Darstellung aller Messdaten .....	122
E	Berechnungsbeispiel WCT1 .....	127
F	Blockschaltbilder und Gleichungen der evaluierten Netze.....	133
<b>LITERATUR.....</b>		<b>136</b>

## Abbildungsverzeichnis

Abbildung 1.1:	Glasproduktion in Deutschland, 2012.....	2
Abbildung 1.2:	Längsschnitt einer exemplarischen Floatanlage.....	3
Abbildung 1.3:	Typische regenerative Querbrennerwanne für Flachglas .....	3
Abbildung 1.4:	Hauptströmungen beim Glasschmelzprozess .....	4
Abbildung 1.5:	Verallgemeinerter Temperaturregelkreis mit verschiedenen Varianten .....	9
Abbildung 2.1:	Strukturbild eines zweischichtigen NARX-Netzes.....	17
Abbildung 2.2:	Grundlegende Architekturen für dynamische Netze .....	18
Abbildung 2.3:	Signalflussplan eines zweischichtigen NARX-Netzes mit drei Eingangsgrößen, fünf verborgenen Neuronen, vier TDL, zwei Ausgangsgrößen und "closed loop"-Architektur .....	19
Abbildung 2.4:	Vereinfachte Übersicht der Messanordnung.....	22
Abbildung 2.5:	Zeitlicher Verlauf von BATCH im gesamten Messzeitraum .....	24
Abbildung 2.6:	Sprunghafte Änderungen von BATCH, GFC und CAF .....	25
Abbildung 2.7:	Gegenüberstellung der Einzelgasmengen GF1, GF2 und GF4 .....	26
Abbildung 2.8:	Gegenüberstellung der Gasmengen und Temperaturen im Schmelzbereich .....	27
Abbildung 2.9:	Zeitliche Verläufe der Prozessgrößen in der Abstehtwanne .....	29
Abbildung 2.10:	Zeitliche Verläufe von GLASS und WEP .....	31
Abbildung 2.11:	Aufbereitung und Erweiterung der Messzeitpunkte .....	34
Abbildung 2.12:	Lineares und konstantes Auffüllen des Datensatzes .....	35
Abbildung 2.13:	Informationsverlust durch gleichmäßige Verteilung der Messzeitpunkte .....	35
Abbildung 2.14:	Vergleich von originalen und vereinfachten Messdaten.....	36
Abbildung 2.15:	Grafische Benutzeroberfläche zur Konfiguration neuronaler Netze .....	37
Abbildung 2.16:	Beispielhaftes Ausgabediagramm des Konfigurationsassistenten.....	39
Abbildung 3.1:	Vergleich zwischen der einzeln (MISO) und gleichzeitig (MIMO) berechneten Ausgangsgröße WCT2 .....	42
Abbildung 3.2:	Testergebnis von Netz G zur Berechnung des Glasstandes .....	43
Abbildung 3.3:	Trainings- und Testergebnisse vom NAR-Netz I .....	44
Abbildung 3.4:	Übersicht der fünf evaluierten Netze .....	45



---

Abbildung 3.5:	Zeitliche Verläufe der tatsächlichen und berechneten Ausgangsgröße WCT2 im Testzeitraum für verschiedene Trainingsmethoden .....	46
Abbildung 3.6:	Simulationsergebnisse von Netz 1 (WCT2) im Trainings- und Testzeitraum für verschiedene Aktivierungsfunktionen.....	49
Abbildung 3.7:	Vergleich der Trainings- und Testergebnisse der Netze 1 und 3 bei Verwendung der Aktivierungsfunktion <i>satlins</i> .....	50
Abbildung 3.8:	Standardabweichungen in Abhängigkeit der Neuronenanzahl bei linearer Aktivierungsfunktion .....	52
Abbildung 3.9:	Normierte Standardabweichungen in Abhängigkeit der Neuronenanzahl bei sigmoider Aktivierungsfunktion .....	53
Abbildung 3.10:	Vergleich der Testergebnisse für WCT2 bei verschiedenen Neuronenzahlen und Aktivierungsfunktionen .....	54
Abbildung 3.11:	Trainingszeiten der Netze 1 und 3 in Abhängigkeit der Neuronenanzahl und der Aktivierungsfunktion .....	54
Abbildung 3.12:	Normierte Trainingszeiten in Abhängigkeit der Neuronenanzahl .....	55
Abbildung 3.13:	Standardabweichungen und Trendlinien in Abhängigkeit der TDL .....	56
Abbildung 3.14:	Zusammenfassende Gegenüberstellung der Testergebnisse von Netz 1 und 4 für verschiedene TDL und Aktivierungsfunktionen.....	57
Abbildung 3.15:	Trainingszeiten und Trendlinien in Abhängigkeit der TDL.....	58
Abbildung 3.16:	Allgemeine Tendenz der Standardabweichungen in Abhängigkeit der Länge vom Trainings- und Testzeitraum .....	60
Abbildung 3.17:	Standardabweichungen und Spannweiten von Netz 1 mit 120 TDL bei zehn Simulationsdurchläufen.....	62
Abbildung 3.18:	Streudiagramm zur Bewertung der Reproduzierbarkeit bei Änderung ausgewählter Netzparameter .....	63
Abbildung 3.19:	Gegenüberstellung der berechneten Standardabweichungen und benötigten Trainingszeiten der Netze 1 und 4 mit konstanter und zufälliger Initialisierung .....	65
Abbildung 3.20:	Testergebnisse der Netze 1 und 3 für verschiedene Konfigurationen bei Verwendung des vereinfachten Datensatzes .....	67
Abbildung 4.1:	Unterschied zwischen OSP und MSP am Beispiel der Netze 1 und 5 .....	71
Abbildung 4.2:	Prädiktionsergebnisse zum Einfluss der Neuronenanzahl .....	72
Abbildung 4.3:	Prädiktionsergebnisse von Netz 1 und 3 zum Einfluss der TDL.....	73
Abbildung 4.4:	Vergleich der Prädiktionsergebnisse für OSP und MSP bei optimaler Konfiguration von Netz 1 .....	74
Abbildung 4.5:	Prädiktionsergebnisse von Netz 1 in Abhängigkeit der Länge des Trainingszeitraums.....	75

---

Abbildung 4.6:	Abhängigkeit der Standardabweichungen bei verschiedenen Trainings- und Prädiktionshorizonten .....	76
Abbildung 4.7:	Vergleich einer sechsstündigen mit einer stündlich neu durchgeführten Prädiktion am Beispiel von Netz 1 .....	77
Abbildung 5.1:	Verwendungsmöglichkeiten von KNN für Regelungszwecke.....	80
Abbildung 5.2:	Grundprinzip der neuronalen prädiktiven Regelung.....	86
Abbildung 5.3:	Blockschaltbild für ein Neuron mit $R$ Eingangsgrößen.....	87
Abbildung 5.4:	Blockschaltbild eines zweischichtigen Netzes mit $R$ Eingängen, $S^1$ Neuronen und einem Ausgang.....	88
Abbildung 5.5:	Blockschaltbild eines zweischichtigen NIO-Netzes mit $R$ Eingangsgrößen, $S^1$ Neuronen, einer Ausgangsgröße und TDL zurückliegenden Messwerten .....	89
Abbildung 5.6:	Vereinfachtes Blockschaltbild eines zweischichtigen NARX-Netzes mit $R$ Eingangsgrößen, $S^1$ Neuronen, einer Ausgangsgröße und TDL zurückliegenden Messwerten .....	90
Abbildung 5.7:	Grundprinzip der OSP für ein NARX-Netz mit zehn TDL .....	91
Abbildung 5.8:	Grundprinzip der Multi Step Prediction am Beispiel eines NARX-Netzes mit TDL = 10 und $N_u = 2$ .....	92
Abbildung 5.9:	Beispiel der zu berechnenden partiellen Ableitungen nach Gleichung (5.35) .....	100
Abbildung 5.10:	Ablaufplan und Berechnungsanleitung zum Newton-basierten Levenberg-Marquardt-Algorithmus zur Berechnung optimaler Stellgrößen .....	104

## Tabellenverzeichnis

Tabelle 1.1:	Bewertung ausgewählter Modellierungsmöglichkeiten bezüglich der Prozesskompatibilität und Eignung zur Regelung .....	12
Tabelle 2.1:	Auflistung der verfügbaren Messgrößen.....	23
Tabelle 2.2:	Kennzahlen zur Beurteilung der Genauigkeit .....	38
Tabelle 3.1:	Simulationsumgebung und Testreihenfolge .....	41
Tabelle 3.2:	Ranking der verschiedenen Trainingsmethoden .....	47
Tabelle 3.3:	Kombinationsmatrix der Aktivierungsfunktionen, Testergebnisse für Netz 3 .....	51
Tabelle 3.4:	Vergleich der Standardabweichungen im Testzeitraum mit den Aktivierungsfunktionen <i>purelin</i> und <i>tansig</i> und den TDL-Werten 10 und 120.....	57
Tabelle 3.5:	Durchschnittliche Standardabweichungen in Abhängigkeit der Trainings- und Testdauer .....	59
Tabelle 3.6:	Untersuchte Netzkonfigurationen zur Bewertung der Reproduzierbarkeit.....	61
Tabelle 3.7:	Vergleich der Standardabweichungen und Trainingszeiten bei verschiedenen Konfigurationen der TDL und Neuronen .....	67
Tabelle 3.8:	Übersicht der evaluierten Netzkonfigurationen und Testergebnisse .....	69
Tabelle 4.1:	Vergleich zwischen OSP und MSP für Netz 1 im dargestellten Prädiktionszeitraum.....	74
Tabelle 5.1:	Abgleich bestehender Regelstrukturen mit gestellten Anforderungen zur Bewertung der Prozesskompatibilität .....	84
Tabelle A.1:	Verfügbare Trainingsfunktionen in <i>MATLAB</i> .....	115
Tabelle B.1:	Verfügbare Aktivierungsfunktionen in <i>MATLAB</i> .....	117
Tabelle C.1:	Umrechnungen in das SI-Einheitensystem.....	120
Tabelle C.2:	Kennzahlen aller verfügbaren Prozessgrößen im gesamten Messzeitraum.....	121
Tabelle E.1:	Statistische Kennzahlen der Wichtungsmatrizen und Schwellwerte bei verschiedener Aktivierungsfunktion und Neuronenanzahl zur Berechnung der Ausgangsgröße WCT1.....	132

## Verzeichnis der Symbole und Abkürzungen

### Abkürzungen / Akronyme

ACSI	Advanced Control Solutions Inc.
CFD	Computational Fluid Dynamics
DIC	Direct Inverse Control
FC	Feedforward Control
FL	Feedback Linearization
FLC	Fuzzy Logic Control
HVG	Hüttentechnische Vereinigung der Deutschen Glasindustrie e.V.
IMC	Internal Model Control
KNN	Künstliches Neuronales Netz
LM	Algorithmus von Levenberg-Marquardt
MIMO	Multiple Input Multiple Output
MISO	Multiple Input Single Output
MPC	Model Predictive Control
MSE	Mean Squared Error
MSP	Multi Step Prediction
NAR	Nonlinear AutoRegressive
NARX	Nonlinear AutoRegressive with EXternal (Exogenous) Input
NIO	Nonlinear Input Output
OMC	Optic Melting Control
OSP	One Step Prediction
POD	Proper Orthogonal Decomposition
SISO	Single Input Single Output
STG	Software & Technologie Glas GmbH
TDL	Tapped Delay Lines
TNO	Niederländische Organisation für Angewandte Naturwissenschaftliche Forschung

### Formelzeichen und Operatoren

$a$	Neuronenausgangswert, -vektor
$a_{norm}$	normierter Neuronenausgangswert, -vektor
$b$	Schwellwert, -vektor

$e$	Fehler, Regelabweichung
$e_{\max,+}$ , $e_{\max,-}$	positiver, negativer maximaler Fehler
$\bar{e}$	arithmetischer Fehler-Mittelwert
$E$	Vektor zukünftiger Regelabweichungen, Vektor der Fehler
$\delta$	Toleranz
$f$	Suchrichtung
$G$	Gradient, Gütefunktion
$H$	Hesse-Matrix
$iw$ , $IW$	Wichtungsfaktor, Wichtungsmatrix für Eingangsgrößen
$J$	Gütefunktion
$\lambda$	Levenberg-Marquardt-Parameter
$lw$ , $LW$	Wichtungsfaktor, Wichtungsmatrix für Ausgangsgrößen
$\mu$	Schrittweite
$n$	Neuroneneingangswert, -vektor
$n_{norm}$	normierter Neuroneneingangswert, -vektor
$\tilde{n}$	Neuroneneingangswert, abhängig von zukünftigen Ein- und Ausgangsgrößen
$N_u$	Stellhorizont
$N_1$	Beginn vom Prädiktionszeitraum
$N_2$	Ende vom Prädiktionszeitraum
$r$	Sollwert, Residuum
$R$	Anzahl Eingangsgrößen, Sollwertvektor, Matrix im Gauss-Newton-Verfahren
$\rho$	Wichtungsfaktor Stellaufwand
$S$	Standardabweichung
$S^1$ , $S^2$	Neuronenanzahl Schicht 1, Schicht 2
$u$	Eingangsgröße, Stellgröße
$u_{norm}$	normierte Eingangsgröße, Stellgröße
$U$	Vektor zukünftiger Stellgrößen
$\Delta u$	Stellgrößenänderung
$x$	Operator beim Gaußschen Eliminationsverfahren
$y$	Ausgangsgröße, Regelgröße
$y_{mess}$	gemessene Ausgangsgröße, Regelgröße
$y_{norm}$	normierte Ausgangsgröße, Regelgröße
$\hat{y}$	Vorhersagewert der Ausgangsgröße
$\hat{Y}$	Vektor der vorhergesagten Ausgangsgrößen

Indizes

$d$	bezogen auf den TDL-Zeitraum
$i$	bezogen auf die Iteration
$j$	bezogen auf das Neuron in der verborgenen Schicht
$k$	bezogen auf den Zeitpunkt im Prädiktionszeitraum
$l$	bezogen auf den Zeitpunkt im Stellhorizont $N_u$

---

# 1 Problemstellung und Zielsetzung

Glas ist ein faszinierender Werkstoff, der durch seine vielfältigen Anwendungsmöglichkeiten aus dem heutigen Leben nicht mehr wegzudenken ist. Die industrielle Glasherstellung geht bis Mitte des 18. Jahrhunderts zurück und stellt heute einen komplexen und anspruchsvollen Prozess dar. Heutige Flachglasanlagen mit Längen von mehreren hundert Metern zeugen von einer mittlerweile vollständig automatisierten Produktionskette. Die Herstellung von Glas ist ein sensibler und interaktiver Prozess. Die Qualität des Endproduktes ist im hohen Maße von der Qualität der Glasschmelze abhängig. Aus diesem Grund haben verbesserte Regelungs- und Automatisierungstechniken für den Schmelzprozess in der Praxis zunehmende Bedeutung erlangt.

Die wohl wichtigste Prozessgröße bei der Glasherstellung ist die Temperatur. Um eine hohe Produktqualität zu gewährleisten, ist die strenge Einhaltung vorgegebener Temperaturniveaus unabdingbar. Aufgrund der schwierigen Bedingungen im Schmelzofen entwickelten sich unterschiedliche Herangehensweisen, um den Schmelzprozess zu optimieren. Neben konventionellen einschleifigen Temperatur- und Leistungsregelkreisen wurden erweiterte Regelstrategien integriert. Im Fokus wissenschaftlicher Forschungen stehen derzeit modellbasierte prädiktive Ansätze. Ein weiterer Forschungsschwerpunkt ist die Analyse und Optimierung anhand numerischer Strömungssimulationen.

Bevor im zweiten Abschnitt bestehende Strategien zur Prozessoptimierung beschrieben werden, stellt der erste Abschnitt den Werkstoff Glas und den Herstellungsprozess für Flachglas einleitend vor. Der dritte Abschnitt fasst die regelungstechnischen Problemstellungen zusammen, die zur abgeleiteten Zielsetzung dieser Arbeit führen.

## 1.1 Eigenschaften und Herstellung von Glas

Glas ist ein anorganisches Schmelzprodukt, das beim Abkühlen nicht kristallisiert. Die Flüssigkeitsstruktur der Glasschmelze bleibt auch im festen Zustand erhalten. Die wichtigsten Glasrohstoffe sind Sand, Soda, Kalk und Dolomit. Weitere zusätzliche Komponenten dienen zur Einstellung spezieller Produkteigenschaften. Der Werkstoff Glas zeichnet sich vor allem durch seine chemische Beständigkeit und Durchlässigkeit für das sichtbare Licht aus. Dadurch sind Gläser besonders zum Konservieren und für optische Anwendungen

geeignet. Flachglas wird hauptsächlich für den konstruktiven Bau verwendet und kann u. a. zu Isolier- und Sicherheitsglas weiterverarbeitet werden. Abbildung 1.1 zeigt für das Jahr 2012 die gesamte Jahresproduktion für den Standort Deutschland, aufgeteilt nach Glasart und deren Produktionsmenge und -wert. Flachglas ist mit 27 % das zweitmeiste produzierte Glas. Diese Produktion verteilt sich derzeit auf zehn Flachglasanlagen. Weltweit werden in rund 180 Anlagen ca. 40 Millionen Tonnen Flachglas pro Jahr produziert [1].

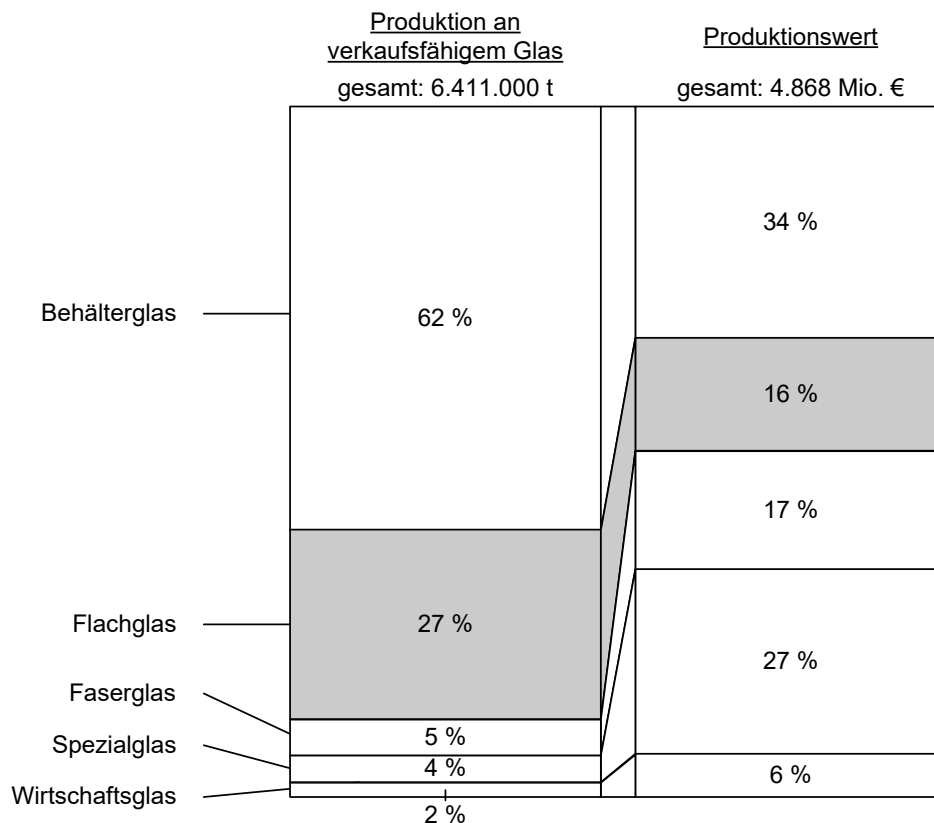


Abbildung 1.1: Glasproduktion in Deutschland, 2012 (Datenbasis: [2])

Der Beginn der industriellen Glasherstellung reicht bis ins Jahr 1867 zurück, als in Dresden die erste Schmelzwanne mit Wärmerückgewinnung über Regeneratoren in Betrieb genommen wurde. Die Industrialisierung der Glasproduktion bekam einen weiteren Schub durch die Entwicklung der Owens-Maschine im Jahr 1903, die auf Basis eines Blasverfahrens die automatische Hohlglasherstellung beschleunigte. Unabhängig von der Glasart besteht der Glasherstellungsprozess aus folgenden Schritten: Rohmaterial (Gemenge) wird über eine automatische Vorrichtung der Schmelzwanne zugeführt, in der es durch Wärmezufuhr schmilzt. Aufgrund der entstehenden Temperaturgradienten entwickeln sich Konvektionsströmungen, die zur Homogenisierung der Glasschmelze führen. Das geschmolzene Glas wird anschließend der Schmelzwanne entnommen und zum jeweiligen Produkt weiterverarbeitet.



Je nach Glasart und Produktanforderungen haben sich verschiedene Herstellungsverfahren entwickelt. Heute wird Flachglas nach zwei Formgebungsverfahren hergestellt. Beim nicht mehr so weit verbreiteten Gusswalzglasverfahren wird am Ende des Schmelzofens ein kontinuierliches Glasband entnommen und zwischen zwei sich gegensinnig drehenden Walzen geformt. Eine höhere Produktqualität kann mit dem kontinuierlichen Floatglasverfahren erreicht werden. Floatanlagen produzieren Flachglas bei einer Tonnage von 450 - 1.000 t/d mit Stärken zwischen 0,5 - 25 mm und einer Breite bis zu 3,5 m [3]. Die Laufzeit einer Flachglaswanne liegt mit 10 - 14 Jahren im gleichen Bereich wie für Behälterglaswannen, trotz der deutlich größeren Bauform. Allein der Schmelzbereich hat bei Floatanlagen eine Länge von 25 - 40 m [4]. Abbildung 1.2 zeigt eine vereinfachte schematische Darstellung einer Floatanlage. Der Schmelz- und Abstehbereich wird in Abbildung 1.3 detaillierter dargestellt.

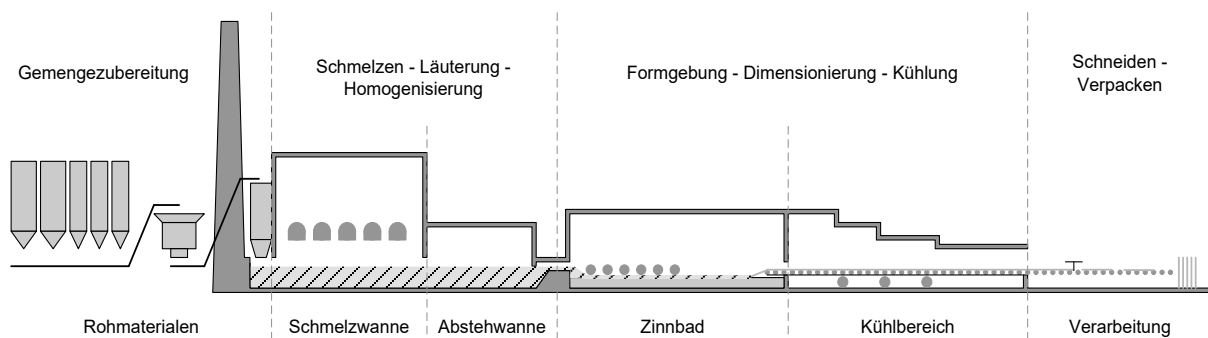


Abbildung 1.2: Längsschnitt einer exemplarischen Floatanlage

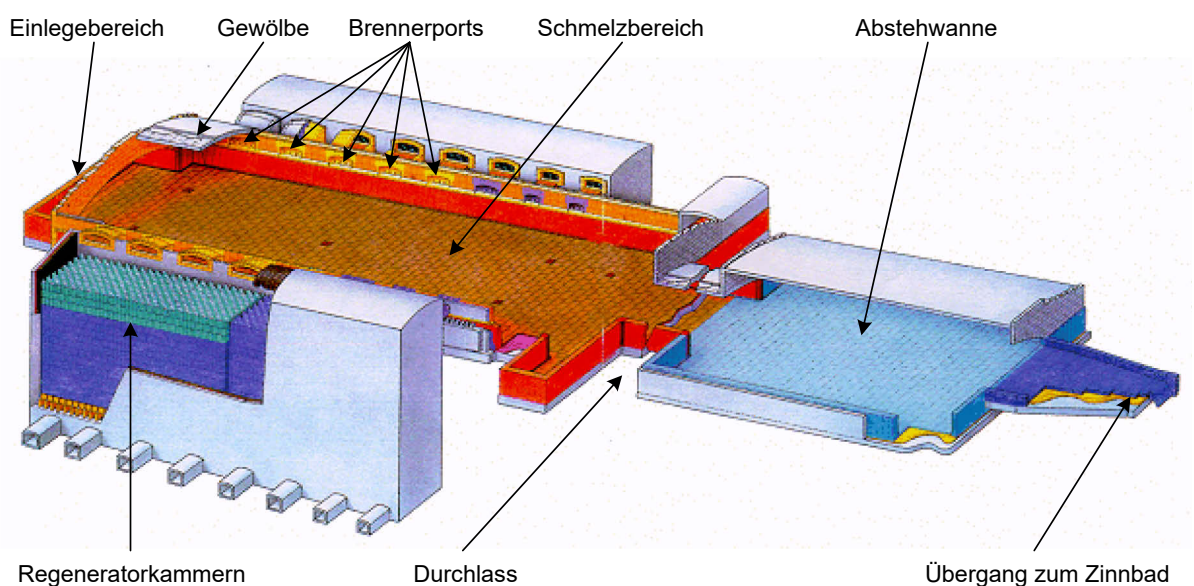


Abbildung 1.3: Typische regenerative Querbrennerwanne für Flachglas [5]

Automatische Gemengezubereitungsanlagen transportieren das Rohstoffgemisch über Laufbänder zum Einlegevorbau, wo das Gemenge über Schieber, meist in Kombination mit einem Rüttlertisch, direkt auf das Glasbad geschoben wird. Die chemische Zusammensetzung, die Korngröße, der Wassergehalt und das Verstaubungsverhalten des Gemenges beeinflussen den nachfolgenden Schmelzprozess und müssen wenn möglich gemessen und kontrolliert werden. Für Flachglas werden überwiegend mit Erdgas betriebene regenerative Querbrennerwannen verwendet (Abbildung 1.3). Die Aufgabe des Schmelzaggregates ist es, mit möglichst geringem Energieeinsatz eine homogene Schmelze für den weiteren Verarbeitungsprozess bereitzustellen. Die Schmelztemperatur liegt bei über 1.500 °C. Um den Energieverbrauch zu reduzieren wird auf Wärmerückgewinnung gesetzt. Die beim Schmelzprozess entstehenden Gase (z. B. Abgase der Verbrennung, freiwerdende Gemengegase) werden über separate Schächte in Regeneratorkammern abgeführt. Die vorhandene Wärme der Abgase wird in einer keramischen Gitterung gespeichert, mit der die Verbrennungsluft auf bis zu 1.400 °C vorgewärmt werden kann. Auf jeder Querseite stehen mehrere Brennerports für die Wärmezufuhr zur Verfügung, die wechselseitig in gleichmäßigen Abständen (15 - 20 min) feuern. Die dabei eingesetzte Brennertechnik hat Einfluss auf die Flammenlage, -länge, -temperatur und die Schadstoffemissionen.

Hinsichtlich eines guten Schmelzergebnisses spielen zusätzlich die Vorgänge im Glasbad eine besondere Rolle. Laut [6] werden ca. 75 % des Gemenges allein durch die Energie der Glasschmelze geschmolzen. Nur 25 - 40 % der von der Flamme thermisch zugeführten Energie wirkt direkt auf die Schmelze. Durch die unterschiedlichen Temperaturgradienten im vorderen Schmelzbereich entstehen Konvektionsströmungen, die die Homogenisierung der Glasschmelze und letztlich die Glasqualität wesentlich beeinflussen (Abbildung 1.4).

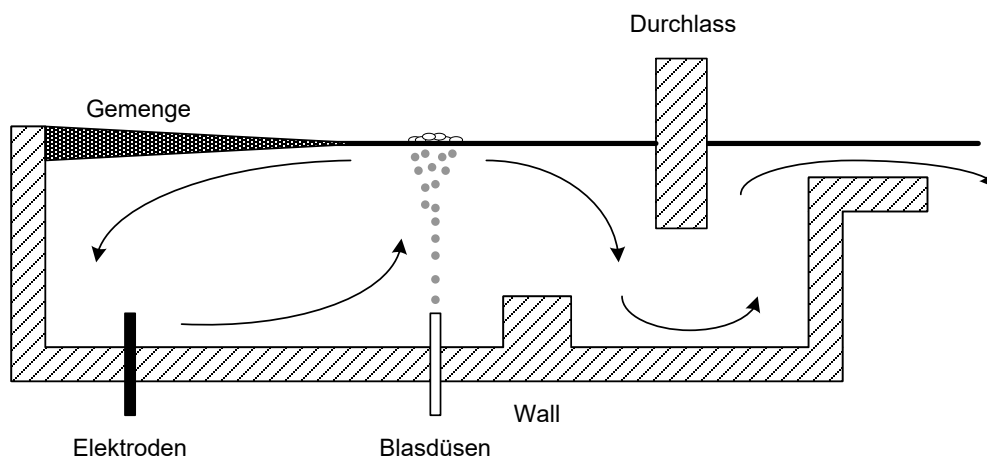


Abbildung 1.4: Hauptströmungen beim Glasschmelzprozess

Durch den Einbau von Strömungshilfen kann die Homogenisierung optimiert werden. Elektroden im Bodenbereich sorgen für eine zusätzliche Energiezufuhr. Eingebaute Blasdüsen setzen Läutermittel frei, die bei den vorherrschenden Temperaturen verstärkt zur Gasbildung neigen. Die aufwärts strömenden Blasen vermengen sich mit kleineren in der Glasschmelze vorhandenen Blasen und werden durch den Auftrieb aus der Schmelze transportiert.

Ein im Boden eingebauter Wall und der sogenannte Durchlass sind weitere konstruktive Maßnahmen, mit denen die Strömungsvorgänge im Glasbad beeinflusst werden können. Der Durchlass stellt in erster Linie eine räumliche Trennung zwischen der Schmelzwanne und der Abstehtwanne dar. Ist er wie in Abbildung 1.4 ins Glasbad eingelassen, hat dies zusätzlich den Vorteil, dass Verunreinigungen an der Glasbadoberfläche nicht in den Abstehtbereich gelangen. Demgegenüber steht ein erhöhter Verschleiß durch die kontinuierlich vorbeiströmende Glasschmelze.

Nach [3] beträgt die Verweildauer der Schmelze allein im Schmelz- und Abstehtbereich bis zu zwölf Stunden. Die laminaren Strömungen erreichen durchschnittlich eine Strömungsgeschwindigkeit von 0,00014 m/s in der Schmelzwanne und 0,002 m/s in der Abstehtwanne [7]. Die Bereitstellung einer gleichmäßig homogenen Glasschmelze hängt demzufolge von dem Zusammenspiel der turbulenten Prozesse oberhalb und den langsamen physikalisch-chemischen Transportvorgängen innerhalb der Glasschmelze ab.

In der Abstehtwanne wird die homogenisierte Glasschmelze auf ca. 1.050 °C kontrolliert gekühlt. Am hinteren Ende der Abstehtwanne fließt die Schmelze über den sogenannten Lippenstein auf das für das Floatglasverfahren typische Zinnbad. Dabei wird der Effekt genutzt, dass sich bei zwei nicht mischbaren Flüssigkeiten die leichtere auf der schwereren Flüssigkeit gleichmäßig ausbreitet. Zinn hat eine höhere Dichte als die Glasschmelze, ist ihr gegenüber chemisch inert, hat eine geringe Schmelztemperatur und verdampft bei den gegebenen Temperaturen nur geringfügig [3]. Über die gesamte Länge des Zinnbades wird das Glasband sukzessive heruntergekühlt. Die Dicke und Breite des Glasbandes wird im vorderen Zinnbadbereich durch wassergekühlte Rollen (sogenannte Top Roller) gesteuert, die direkt am äußeren Rand auf dem Glasband aufliegen und je nach Winkel das Glas strecken oder stauchen. Am Ende des Zinnbades wird das Glasband bei einer Temperatur von ca. 600 °C über Transportrollen der weiteren Verarbeitung zugeführt.

Floatanlagen sind mehrere hundert Meter lang, da besonders der Kühlungsprozess schrittweise erfolgen muss, um thermische Spannungen im Glas gering zu halten. Am Ende jeder Floatanlage wird das Glas veredelt, eingehend auf Qualität untersucht, geschnitten und voll-

automatisch verpackt. Damit steht ein kontinuierliches Produktionsverfahren zur Verfügung, das eine hohe Produktionskapazität sowie einen hohen Grad an Automatisierung erlaubt. In Deutschland werden ca. 95 % aller Flachgläser nach diesem Verfahren hergestellt [3].

### 1.2 Stand der Technik und Forschung

Erste Ansätze zur Regelungs- und Automatisierungstechnik für den Schmelzbereich basierten auf einfachen Temperatur- und Leistungsregelkreisen, u. a. in [8 - 10]. Steigende Qualitätsanforderungen zwangen die Anlagenbetreiber zu einer stetigen Optimierung des Schmelzprozesses. Dabei lassen sich die Entwicklungen grob in zwei Richtungen teilen. Neben einfachen Regelstrategien, die zumeist auf Black-Box-Modellen basieren und auf eine automatische Prozessführung zielen, wurden zeitgleich analytische Prozessmodelle entwickelt, die mit den Erhaltungsgleichungen für Masse und Energie für mehr Verständnis der inneren physikalischen und chemischen Vorgänge, entweder im gesamten Schmelzofen oder im Glasbad selbst, sorgen sollen, u. a. in [11 - 17]. Erst in den letzten zwei Jahrzehnten hielten zunehmend moderne Regelstrategien Einzug in die Glasindustrie. Nachfolgend werden die verschiedenen Entwicklungen chronologisch vorgestellt, gegliedert nach

- analytischen Prozessmodellen zum Prozessverständnis,
- konventionellen Regelstrategien zur einfachen Prozessführung und
- modernen Regelkonzepten zur übergeordneten Prozessoptimierung.

Erste Entwicklungen zu analytischen Prozessmodellen gehen bis ins Jahr 1975 zurück. Die Hüttentechnische Vereinigung der Deutschen Glasindustrie (HVG) stellte ein mathematisches Modell zur Abschätzung des Energiehaushaltes von Glasschmelzwannen vor [18]. Mit diesem Modell konnten Aussagen über den Einfluss des Wannengewölbes, der Wandwärmeverluste, der Flammenlänge und der Brennstoffart auf die spezifische Schmelzleistung des Ofens getroffen werden. Im Jahr 1977 wurden an der Hochschule Aachen Modelle zur Berechnung von zweidimensionalen Konvektionsströmungen im Glasbad [19, 20] und im Jahr 1982 zur Berechnung von Temperatur-, Strömungs- und elektrischen Potentialfeldern [21] vorgestellt. Ende der Achtziger veröffentlichte die Hochschule Weimar ein dreidimensionales analytisches Modell zur Berechnung von Geschwindigkeitsfeldern in einer vollelektrischen Glasschmelzwanne und ein mathematisches Ofenmodell für brennstoffbeheizte Wannen, das eine Kombination aus einem Energie- und Strömungsmodell darstellt [16, 17, 22]. Dadurch konnte erstmals das gesamte energetische Verhalten einer Glasschmelzwanne simuliert werden. Es folgten weitere internationale Veröffentlichungen

zur mathematischen Modellierung von Transportvorgängen im Glasbad, u. a. [23, 24]. In [14, 25] werden Modelle beschrieben, die sich ausschließlich dem Verbrennungsraum und dem Schmelzprozess widmen. Im Jahr 2008 wurde in [15] ein Gesamtofenmodell vorgestellt. Es setzt sich aus Teilmodellen für Verbrennungsraum, Glasbad und Wände zusammen, die über die Wärmestromdichte gekoppelt sind.

Einen großen Anteil am heutigen Verständnis der inneren Prozessvorgänge haben die Entwicklungen an der niederländischen Organisation für Angewandte Naturwissenschaftliche Forschung (TNO) in Eindhoven. Dort entstand zur Jahrtausendwende das Glasofenmodell *GTM-X* [26, 27]. Dieses CFD-Modell (*Computational Fluid Dynamics*) bildet eine komplette Behälterglaswanne ab, mit der sämtliche Strömungs- und Transportvorgänge im Glasbad und im Oberofen simuliert werden können. Gleichzeitig bietet dieses Modell eine Sammlung von geeigneten Lösern. Mit der Zeit wurde es für verschiedene Wannentypen weiterentwickelt und dient heutzutage als "realer Ofen" zur Identifikation von reduzierten Prozessmodellen. Weitere Beispiele zur Reduzierung von Glasofenmodellen finden sich in [28 - 30]. Mit der Methode der orthogonalen Zerlegung (*proper orthogonal decomposition, POD*) konnte in [28] die Anzahl der Gleichungen eines CFD-Glasofenmodells auf 1 % reduziert werden. Die numerische Strömungssimulation entwickelte sich zu einer weit verbreiteten Möglichkeit um Prozesse in Glasschmelzwannen zu analysieren und zu optimieren. Mittlerweile gehören CFD-Modelle zur Grundausrüstung in der Glasindustrie [31].

Die konventionelle Regelungstechnik hielt schon weit vor Beginn der CFD-Modellierung Einzug in den Glasherstellungsprozess. Eine ausführliche Zusammenfassung an ausgewählten Steuerungs- und Regelungsbeispielen wird in [8] bereits im Jahr 1978 geliefert. Die stets steigenden Anforderungen an Produkt und Umwelt sorgten für einen ständigen Bedarf an Weiterentwicklungen. Die Sorg GmbH, ein deutscher Lieferant von Glasschmelzöfen und -konditionierungssystemen, stellte im Jahr 1996 ein Verfahren zur Regelung der Beheizung von Glas-Wannenöfen vor, mit dem der Energieverbrauch um 5 - 10 % gesenkt werden konnte [10]. Der Ansatz dieses Verfahrens lag in der Berechnung der Wärmestromdichte, die durch den Schmelzspiegel fließt. Die Software & Technologie Glas GmbH (STG) aus Cottbus entwickelte ebenfalls eine Reihe regelungstechnischer Lösungen zur Optimierung des Glasherstellungsprozesses. In verschiedenen Patentschriften wurden Verfahren zur Regelung der Oberofenbeheizung [9], zur Regelung der Flammenlänge [32], zur Bestimmung des Rezirkulationsstroms im Glasbad [33] und zum geregelten Betrieb eines regenerativ beheizten Industrieofens [34] vorgestellt. Bei letzterem wird ein äußerer Regelkreis für die Gewölbetemperatur genutzt, die mit der nötigen Brennstoffmenge geregelt wird. Ein innerer Regelkreis zielt auf eine gleichmäßige Verteilung der Temperaturen in den

Regeneratoren ab. Im Laufe des Betriebs kommt es hier zu unsymmetrischen Temperaturverteilungen, die die Luftvorwärmung der Verbrennungsluft beeinflussen. Als Stellgröße dient eine Änderung der Feuerperiode. Im Jahr 2012 stellte die STG ihr aktuelles Konzept vor, das sich als eine Weiterentwicklung der bisherigen Ansätze versteht. Ein Bildverarbeitungssystem (*Optic Melting Control, OMC*) soll Gemengeinseln, Blasen und Schaumstrukturen erkennen und die Daten in einem Regelkreis zur Schmelzgutbedeckung integrieren [35].

Ein zwar nicht direkt auf den Schmelzbereich bezogenes, dennoch aus konzeptioneller Sicht interessantes Regelkonzept wird in [36] vorgestellt. Zur Temperaturregelung in einem Feeder (Kanal der Glasschmelze bei Behälterglaswannen) wird ein Zonenmodell verwendet. Die Temperatur einer Zone wirkt als Störgröße auf die benachbarte Zone. Das Übertragungsverhalten jeder Zone wurde anhand von aufgezeichneten Sprungantworten nach Änderungen der Brenngaszufuhr und Stellung der Lüftungsklappen identifiziert. Die daraus dimensionierten PID-Reglerparameter führen zu einer verbesserten Temperaturregelung.

Die Weiterentwicklung moderner regelungstechnischer Theorien (Advanced Control) hielt auch in der Glasindustrie zunehmend Einzug. Im Jahr 1997 stellte Siemens ein Verfahren zur Regelung von Brennern einer Glasschmelzwanne auf Basis einer Fuzzy-Logik vor [37]. Adaptive Methoden und Zustandsregelungen wurden u. a. in [38, 39] beschrieben. Einen tieferen Einblick in glastechnische Prozessmodelle und deren Anwendungsmöglichkeiten in erweiterten Regelstrategien gab [40] im Jahr 2002. Die tschechische Firma Glass Service Inc. entwickelte eine Komplettlösung für den gesamten Schmelzprozess, das *Expert System* [41]. Als Basis diente das über Jahre entwickelte CFD-Glasofenmodell *GS-GFM* [42]. Das Regelkonzept zielt dabei nicht auf einen optimierten Prozess an sich, sondern auf die übergeordnete Steuerung einer großen Gruppe von Einzelregelungen. Das *Expert System* gibt es heute in seiner dritten Version und nutzt dabei zwei grundlegende mathematische Algorithmen: Model Predictive Control (MPC) und Fuzzy Logic Control (FLC). Ein ähnlicher Ansatz wurde in [43] verfolgt. Für die MPC-Regelung diente ein vom *GTM-X* berechnetes reduziertes Testmodell als Regelstrecke. Eine weitere MPC-Komplettlösung ist das System *BrainWave* der amerikanischen Firma Advanced Control Solutions Inc. (ACSI) [44 - 47].

Das Entwicklungs- und Optimierungspotential ist längst nicht ausgeschöpft. Anlagenbetreiber stehen aufgrund der inneren, nicht immer verständlichen dynamischen Prozesse bis heute vor Qualitätsmängeln oder Produktionsausfällen. Die meisten Flachglasanlagen arbeiten auch heute noch mit einfachen PID-Temperaturregelkreisen für einzelne Zonen, die bei Unregelmäßigkeiten durch manuelles Eingreifen der Wannenführer korrigiert werden müssen (Abbildung 1.5) [36, 42, 48, 49].

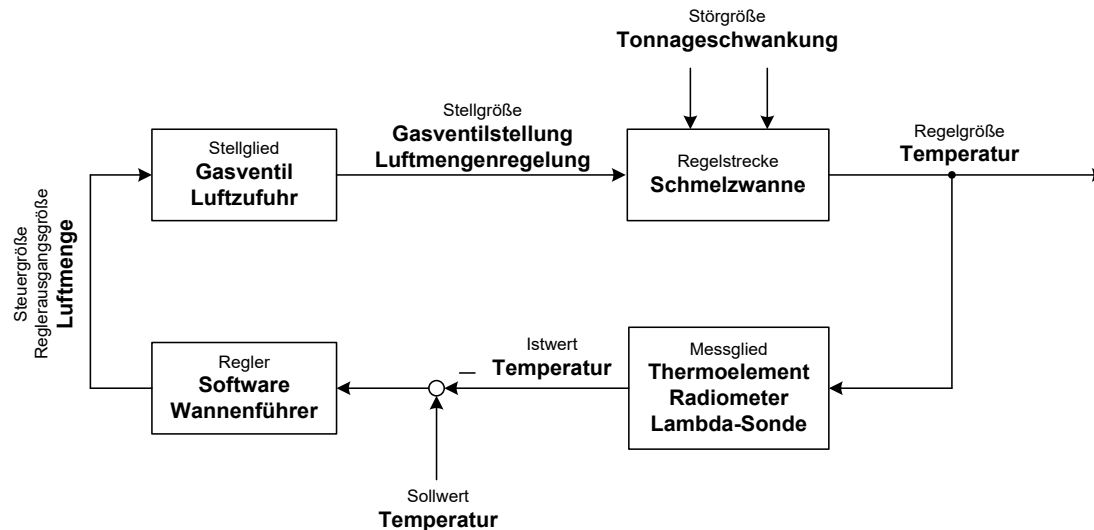


Abbildung 1.5: Verallgemeinerter Temperaturregelkreis mit verschiedenen Varianten [48]

### 1.3 Regelungstechnische Problemstellungen

Die Glasherstellung ist ein sehr langsamer und kontinuierlicher Prozess, der aufgrund der mehrjährigen Ofenlaufzeiten nur wenigen geometrischen und verfahrenstechnischen Änderungen unterworfen ist. Im Jahr 2000 wurden trotz damals bereits bekannter moderner Regelstrategien weltweit schätzungsweise 90 % aller Schmelzwannen weiterhin mit einfachen PID-Temperaturregelkreisen gefahren [41]. Die Tendenz zu Advanced Control ist steigend, dennoch kommt es immer wieder zu unerwarteten Produktionsschwankungen und Qualitätsmängeln. Die Ursachen liegen am thermochemischen Prozess selbst und an verfahrenstechnischen und produktionsbedingten Problemen.

Chemische Prozesse charakterisieren sich in der Regel durch nichtlineares, zeitvariantes Verhalten mit verteilten Parametern und Mehrgrößencharakter [36]. Die einzelnen Prozessgrößen werden von mehreren anderen zeitabhängigen Prozessgrößen beeinflusst. Zudem sind die verfügbaren Stellglieder in der Schmelzwanne in der Regel nichtlinear. PID-Regelungen kommen aufgrund der Totzeiten und mehrstündigen Zeitverzögerungen an ihre Grenzen. Um dem entgegen zu wirken wird oftmals versucht, schnellere und leichter messbare Größen, z. B. die Gewölbetemperatur, zu regeln. Die Gewölbetemperatur weicht jedoch wesentlich von der qualitätsbestimmenden Schmelzbadtemperatur ab.

Ein weiteres Problem im Schmelzbereich ist die Neigung des Gemenges zur Schaumbildung. Diese hängt mit der chemischen Zusammensetzung des Gemenges und dem Anteil

der verwendeten Altglasscherben zusammen. Der isolierend wirkende Schaum reflektiert einen Teil der zugeführten Wärme. Das führt zu steigenden Oberofentemperaturen und somit zu einer Reduzierung der Brennstoffmenge in dieser Zone. Die STG hat dieses Problem erkannt und ein Verfahren zur optischen Erkennung von Schaumstrukturen entwickelt [35]. Die Altglasscherben spielen noch eine weitere wichtige Rolle. Die Art, die Menge und der Zustand, bezogen auf die Korngröße, Verunreinigung und Feuchtigkeit, haben wesentlichen Einfluss auf die Viskosität der Glasschmelze [50]. Die Viskosität ist temperaturabhängig und führt zu unterschiedlichen Strömungsgeschwindigkeiten im Glasbad. Geringe Änderungen der chemischen Zusammensetzung des Gemenges und der Glasschmelze im vorderen Schmelzbereich können zu Schwankungen der Viskosität führen, deren Ursachen vielfältig und weitgehend unkontrollierbar sind. Eine Untersuchung des Strömungsverhaltens in Behälterglaswannen mit Tracerversuchen hat ergeben, dass die Viskosität an den kälteren Außenwänden höher ist und dadurch unterschiedliche und zeitabhängige Viskositätsfelder innerhalb der Glasschmelze entstehen [51]. Ein weiteres Ergebnis der Tracerversuche war es, dass die Eindringtiefe der Wärmestrahlung bei Farbgläsern geringer ist als bei Weißglas. Damit konnte gezeigt werden, dass das Strömungsverhalten der Schmelze zusätzlich von der Glasfarbe abhängig ist.

Neben diesen thermochemischen Problemstellungen gibt es noch verfahrenstechnische Hürden, mit denen sich die Automatisierungs- und Regelungstechnik zu beschäftigen hat. In erster Linie ist die messtechnische Erfassung der qualitätsbestimmenden Parameter eine große Herausforderung. Berührungslose Temperaturmesseinrichtungen, z. B. Pyrometer, haben besonders im Schmelzbereich durch Verstaubungen und Schaumbildung eine eingeschränkte Genauigkeit. Indirekte Messmethoden am Wannenboden sind mit weiteren Zeitverzögerungen verbunden. Oftmals sind die wenigen Temperaturmessstellen nicht repräsentativ genug. Aufgrund der geringen Anzahl an möglichen Stelleingriffen können sich Temperaturschwankungen durch den gesamten Herstellungsprozess fortpflanzen.

Ein bisher wenig beachtetes Problem sind Gasbeschaffenheitsschwankungen. Die HVG hat im Jahr 2013 in einer Umfrage festgestellt, dass bei 10 von 20 Glasherstellern unerwartete, nicht angekündigte Änderungen der Brenngaszusammensetzung zu Problemen bei der Glasherstellung geführt haben [52]. Eine sprunghafte Änderung der chemischen Zusammensetzung des Erdgases, welches innerhalb des Normbereichs liegt, kann zu Änderungen der Flammenlänge und -form und damit zu Veränderungen der Wärmeübertragung auf die Glasschmelze führen. Weiterhin können erhöhte Strömungsgeschwindigkeiten, erhöhte Wandtemperaturen und Erhöhungen der Schadstoffemissionen auf Gasbeschaffenheitsschwankungen zurückgeführt werden [53].



Weitere Störgrößen sind z. B. der im laufenden Betrieb nicht messbare Verschleiß des feuerfesten Materials, Leckagen und dadurch bedingte Falschlufteinbrüche sowie Verschmutzungen und Verunreinigungen des Gemenges und der Altglasscherben [49].

## **1.4 Zielsetzung und Vorgehensweise**

### **1.4.1 Ableitung der Zielsetzung**

Grundlage für den Entwurf einer modernen Regelstrategie ist das hinterlegte Prozessmodell. Primäres Ziel dieser Arbeit ist die Identifikation einer geeigneten Modellstruktur und -parametrierung. Die in Abschnitt 1.1 und 1.3 beschriebenen Prozesseigenschaften und Herausforderungen lassen sich zu Anforderungen formulieren, die an das Prozessmodell gestellt werden. Demgegenüber stehen die in Abschnitt 1.2 beschriebenen Ansätze zur Prozessmodellierung. In Tabelle 1.1 (folgende Seite) erfolgt eine Bewertung hinsichtlich der Kompatibilität ausgewählter Prozessmodelle.

Die herkömmlichen E/A-Modelle sind in der Regel auf einen engen Arbeitsbereich begrenzt und nicht für Mehrgrößensysteme geeignet. Zur Abbildung der vorherrschenden Prozesseigenschaften sind analytische Modelle prädestiniert. Diese Modelle sind auf die konkrete Geometrie eines Ofens bezogen und deshalb nur begrenzt aussagefähig. Zudem hängt die Qualität der Ergebnisse im hohen Maße von der Qualität der Randbedingungen ab.

Die CFD-Modelle erfordern sehr genaue Informationen über den Prozess und geben dennoch nur ein idealisiertes Abbild. Demzufolge können diese Modelle auch nur mit den gegebenen Einschränkungen aussagekräftige Ergebnisse liefern. Der Aufwand zur Konstruktion der CFD-Gitter und die Rechenzeiten sind enorm. Typische, in der Glasindustrie zum Einsatz kommende CFD-Modelle bestehen aus  $10^3$  -  $10^8$  Gleichungen, die zu jedem Zeitschritt gelöst werden müssen [28]. Reduzierte CFD-Modelle werden mittlerweile zur Echtzeitregelung herangezogen, jedoch ist die Modellgenauigkeit eingeschränkt. In [30] wird die durchschnittliche Abweichung reduzierter CFD-Modelle im Vergleich zum ursprünglichen Modell auf 10 % beziffert.

Tabelle 1.1: Bewertung ausgewählter Modellierungsmöglichkeiten bezüglich der Prozesskompatibilität und Eignung zur Regelung

<u>Bedeutung der Symbole:</u> ++ = Anforderung wird sehr gut erfüllt + = Anforderung wird gut erfüllt o = vom Anwendungsfall abhängig - = Anforderung wird kaum erfüllt -- = Anforderung wird nicht erfüllt	Prozessmodelle						
	E/A-Modelle ("Black Box")				Analytische Modelle		
	modellbasiert		wissensbasiert				
	Sprungantwort	Übertragungsfunktion	Künstliche Neuronale Netze	Fuzzy-Logik	Zustandsmodell	CFD-Modell	Reduzierte CFD-Modelle
<b>Abbildung der Prozesseigenschaften</b>							
Nichtlineares Verhalten	-	o	++	+	+	++	++
Große Zeitkonstanten / Totzeiten	+	+	+	-	o	+	+
Verteilte Parameter	-	-	-	-	+	+	+
Zeitvariantes Verhalten	-	-	-	-	+	+	+
<b>Eignung zur Regelung</b>							
Geringer Modellierungsaufwand	+	+	o	o	--	--	--
Kein detailliertes Prozesswissen notwendig	+	-	+	-	-	--	--
Extrapolationsfähigkeit	-	-	o	-	+	o	o
Akzeptable Rechenzeiten	++	++	o	o	o	--	o
Behandlung von Mehrgrößensystemen	-	-	++	++	+	+	+

Die Tatsache, dass auch heute Wannenführer manuell in den Herstellungsprozess eingreifen, führte letztlich zu der Idee, ein Prozessmodell zu entwickeln, welches auf Erfahrungen und gelerntes Prozesswissen zurückgreifen kann. Als wissensbasierte Modelle können Künstliche Neuronale Netze (KNN) oder Fuzzy-Algorithmen herangezogen werden. Der wesentliche Vorteil der KNN gegenüber der Fuzzy-Logik besteht darin, dass kein Prozesswissen bei der Modellierung erforderlich ist. KNN können allein aus Beobachtung des Prozesses dessen Verhalten annehmen. Zusätzlich können die Netze adaptiv trainiert werden, so dass sie sich fortwährend an veränderte Umgebungsbedingungen anpassen.

Der Modellierungsaufwand, die Extrapolationsfähigkeit und die Rechenzeiten sind stark von der Netzstruktur (verwendeter Netztyp, Aufbau, Konfiguration, Trainingsdaten) abhängig. Allgemeingültige systematische Verfahren für die Identifikation einer optimalen Netztopologie

und -konfiguration existieren nicht, dennoch sind laut [54] KNN mittlerweile der am häufigsten verwendete Ansatz zur Black-Box-Modellierung. Eine regelungstechnische Nutzung von KNN in der Glasindustrie ist nach aktuellem Kenntnisstand nicht bekannt. Das Thema Prädiktion mit KNN ist in anderen Anwendungsbereichen bereits präsent. In [55] wurde untersucht, inwieweit sich KNN zur Prädiktion von chaotischen Zeitreihen eignen. Im Vordergrund stand eine geeignete Auswahl der Netzstruktur. In [56, 57] wurde nachgewiesen, dass neuronale Netze ebenfalls zur Prädiktion von Prozessen mit Hystereseverhalten geeignet sind. Weitere interdisziplinäre Beispiele zur Prädiktion mittels KNN sind

- die Bestimmung des benötigten Frachtvolumens im Güterverkehr [58],
- die Berechnung von Muskelbewegungen nach elektrischer Stimulation [59],
- die Berechnung des maximalen Salzgehaltes eines Flusses [60],
- die Prognose des Trinkwasserverbrauchs [61] oder
- des Fernwärmebedarfs anhand historischer Lastdaten und Wetterprognosen [62].

In [63] und [64] finden sich weitere zahlreiche Anwendungsgebiete. Die jeweils verwendete Netzstruktur ist stark anwendungsbezogen. Die Identifikation einer geeigneten Netzstruktur zur Berechnung von qualitätsbestimmenden Prozessgrößen in einem Flachglasschmelzofen ist somit wesentlicher Schwerpunkt dieser Arbeit. In einem weiteren Schritt soll das identifizierte neuronale Prozessmodell für eine erweiterte Regelstrategie genutzt werden. Der modellbasierte prädiktive Ansatz ist vielversprechend. Ziel dieser Arbeit ist es, ein geeignetes neuronales Streckenmodell zu identifizieren und in eine MPC-Regelung einzubetten, so dass die Vorteile der neuronalen Streckenidentifikation mit den Vorteilen der modellbasierten prädiktiven Regelung kombiniert werden können.

### **1.4.2 Struktur der Arbeit**

Grundlage ist ein Datensatz einer realen Flachglasanlage mit mehreren Ein- und Ausgangsgrößen. In Kapitel 2 wird dieser Datensatz eingehend analysiert und dessen notwendige Aufbereitung beschrieben. Zudem wird eine Auswahl der möglichen neuronalen Netzstrukturen getroffen und deren Realisierung mit der Software *MATLAB* aufgezeigt. Für die Identifikation geeigneter KNN wurde eine grafische Benutzeroberfläche programmiert, die die Konfiguration der zu erstellenden Netze erleichtert.

Der Schwerpunkt dieser Arbeit liegt im Entwurf und in der Optimierung einer geeigneten Netzstruktur für den gegebenen Anwendungsfall. Es wird der Einfluss verschiedener Netz-

parameter auf die Genauigkeit der Netze untersucht. Dazu gehören die Auswahl der Ein- und Ausgangsgrößen, die Anzahl der verborgenen Neuronen, die Anzahl und Auswahl der verwendeten Daten, die Aktivierungsfunktionen und Trainingsmethoden. In Kapitel 3 werden die Simulationsergebnisse und Abhängigkeiten der einzelnen Netzparameter in Bezug auf die Genauigkeit der Netze diskutiert.

KNN eignen sich aufgrund ihrer Approximationsfähigkeit sehr gut für prädiktive Aufgaben. Neben der Vorhersage des zukünftigen Ausgangswertes können die Netze durch eine gezielte Änderung der Netzstruktur auch Vorhersagen über einen längeren Zeitraum treffen. Die Multi Step Prediction sowie Fehlereinflüsse verschiedener Parameter werden ausführlich in Kapitel 4 beschrieben.

Im fünften Kapitel wird ein geeignetes Regelkonzept abgeleitet. Ein Abgleich bestehender Regelstrukturen mit hergeleiteten Anforderungen führt zu einer prädiktiven Regelstrategie. Es werden alle dafür notwendigen Komponenten beschrieben sowie Grenzen und Alternativen diskutiert, bevor im letzten Kapitel die Ergebnisse zusammengefasst und mögliche weiterführende Schritte vorgeschlagen werden.

---

## 2 Entwicklungsumgebung

Zur programmiertechnischen Realisierung wurde die Simulationssoftware *MATLAB* verwendet. *MATLAB* ist eine hochentwickelte objektorientierte Programmiersprache zur Lösung, Visualisierung und Analyse von technischen Aufgabenstellungen. Die Software umfasst neben den grundlegenden Standardbibliotheken noch erweiterbare Funktionsbibliotheken, mit denen nahezu alle ingenieurstechnischen Aufgabenbereiche abgedeckt werden können. Für die Bearbeitung von regelungstechnischen Problemstellungen sind vor allem die *Control System Toolbox*, die *System Identification Toolbox*, die *Model Predictive Control Toolbox* und die *Neural Network Toolbox* geeignet. Letztere bietet für die vorliegende Aufgabenstellung einen vordefinierten Funktionsumfang, um Künstliche Neuronale Netze zu erstellen. Im ersten Abschnitt dieses Kapitels werden der Funktionsumfang und mögliche Konfigurationsmöglichkeiten vorgestellt.

Die Identifikation von geeigneten Prozessmodellen erfolgt mit gemessenen Daten einer Flachglasanlage. Der zweite Abschnitt enthält eine Analyse des vorliegenden Datensatzes und liefert eine Interpretation der Messdaten. Die aufgezeigten Zusammenhänge geben einen ersten Eindruck vom vorliegenden Prozessverhalten. Für die weitere Verwendung wurde der Datensatz unterschiedlich aufbereitet. Welche Schritte dafür notwendig waren und welche Konsequenzen sich für die weitere Betrachtung ergeben, wird ebenfalls in diesem Abschnitt thematisiert.

Die Programmierung eines KNN mit *MATLAB* kann mit wenig Aufwand realisiert werden. Um den Einfluss der verschiedenen Netzparameter untersuchen zu können, sind vor jeder Simulation Änderungen im Quellcode notwendig. Schon bei der Auswahl der Eingangsgrößen sind 1.023 verschiedene Kombinationen möglich. Für diesen Zweck wurde eine grafische Benutzeroberfläche programmiert, die die Konfiguration und Bereitstellung der Messdaten erleichtert und somit eine strukturierte Herangehensweise zulässt. Der Aufbau des Konfigurationsassistenten und die verwendeten Möglichkeiten zur Bewertung der Simulationsergebnisse sind Inhalt des dritten Abschnitts.

## 2.1 Realisierung Künstlicher Neuronaler Netze mit *MATLAB*

### 2.1.1 Funktionsumfang und Anwendungsgebiete

Die Anwendungsmöglichkeiten von KNN sind vielfältig. Heutzutage können mit KNN Trajektorien moderner Industrieroboter optimiert oder Aktienkurse prognostiziert werden. In der Sprach- und Bilderkennung spielen KNN ebenfalls eine große Rolle. Die Software *MATLAB* bietet mit der integrierten *Neural Network Toolbox* ein Werkzeug zur Gestaltung, Implementierung, Visualisierung und Simulation von neuronalen Netzen. Insbesondere folgende Aufgaben können mit dieser Toolbox realisiert werden:

- Fitting a Function (Funktionsapproximation),
- Recognizing Patterns (Mustererkennung),
- Clustering Data (Klassifizierung) und
- Time Series Prediction (Prognose).

Das Anpassen an eine Funktion, die Mustererkennung oder das Gruppieren von ähnlichen Daten sind Problemstellungen, die sich durch rein statisches Verhalten charakterisieren lassen. Das Netz lernt auf Basis verschiedener Eingangsgrößen einen statischen Zusammenhang. Beispiele hierfür sind u. a. die Bestimmung von Grundstückspreisen anhand der Verkehrsanbindung, der Anzahl der Schulen und der Kriminalitätsrate. Die Klassifizierung der Weinsorte auf Basis der chemischen Zusammensetzung ist ein typisches Beispiel für eine Mustererkennung. In der Medizintechnik werden KNN zur Auswertung von MRT-Bildern (Magnetresonanztomographie) genutzt, um Tumore frühzeitig zu erkennen [65]. Die Eingangsgrößen sind weder zeitabhängig, noch gibt es untereinander prozesstechnische Zusammenhänge. Zur Realisierung einer erweiterten Regelstrategie kommen daher nur dynamische Netze infrage, die mit vergangenen Ein- und Ausgangsgrößen zukünftige Ausgangsgrößen berechnen. Dynamische KNN lassen sich in zwei Kategorien einteilen: vorwärtsgerichtet (*feedforward*) und rückgekoppelt (*recurrent*). Im Unterschied zu rückgekoppelten Netzen können Signale bei Feedforward-Netzen nur vom Eingang zum Ausgang verlaufen. Ausführliche Informationen zu Anwendungsgebieten und von *MATLAB* bereitgestellten Netztopologien sind in [66] beschrieben.

### 2.1.2 Grundlegendes Design

Der prinzipielle Aufbau und die Funktionsweise eines einzelnen Neurons sowie deren unterschiedliche Zusammenführung zu bekannten Grundtypen von Künstlichen Neuronalen Netzen finden sich u. a. in [64, 67, 68]. Für die Time Series Prediction stehen drei Netztypen zur Verfügung. Beim NIO-Netz (**N**onlinear **I**nter **O**utput) wird die zukünftige Ausgangsgröße  $y(t)$  lediglich aus den  $d$  vergangenen Werten der Eingangsgröße  $u(t)$  berechnet. Das NAR-Netz (**N**onlinear **A**uto**R**egressive) berechnet die zukünftige Ausgangsgröße  $y(t)$  nur anhand der vergangenen Ausgangsgrößen. Dieser Netztyp wird gewählt, wenn keine messbaren Eingangsgrößen vorliegen. Eine Kombination beider Varianten ist das NARX-Netz (**N**onlinear **A**uto**R**egressive with **E**xternal (Exogenous) Input).

$$\text{NIO:} \quad y(t) = f(u(t-1), \dots, u(t-d)) \quad (2.1)$$

$$\text{NAR:} \quad y(t) = f(y(t-1), \dots, y(t-d)) \quad (2.2)$$

$$\text{NARX:} \quad y(t) = f(y(t-1), \dots, y(t-d), u(t-1), \dots, u(t-d)) \quad (2.3)$$

Abbildung 2.1 zeigt das Strukturbild eines standardmäßigen NARX-Netzes. Es setzt sich aus einem Eingangsvektor  $u(t)$  und zwei nacheinander folgenden Schichten zusammen. Der erste hochgestellte Index steht für die jeweilige Schicht, der zweite Index gibt an, aus welcher Schicht die Signale bezogen werden.

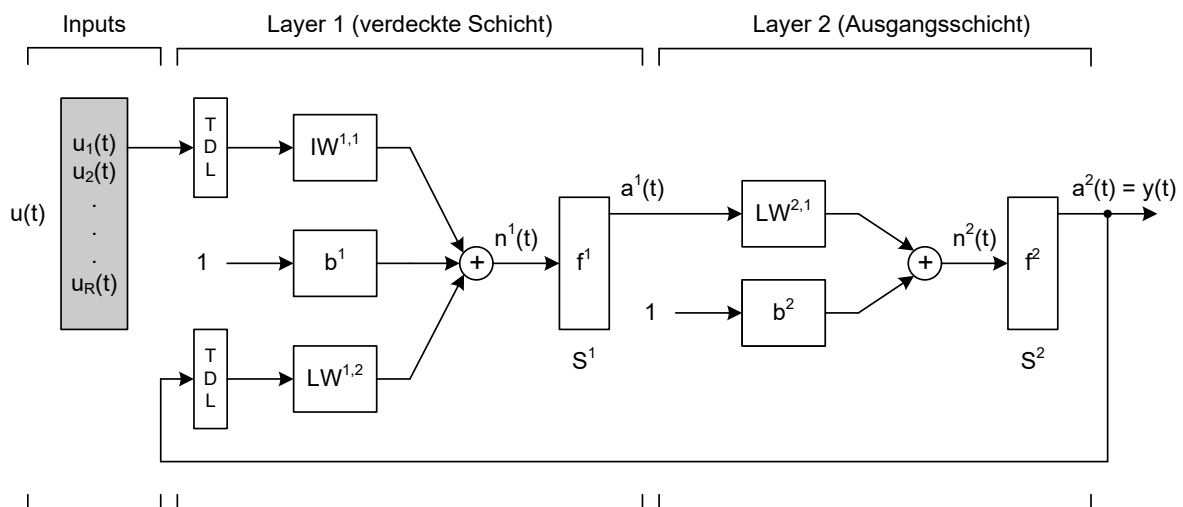


Abbildung 2.1: Strukturbild eines zweischichtigen NARX-Netzes (Basis: [67])

Der Netz-Eingangsvektor  $u(t)$  beinhaltet  $R$  unabhängige Eingangsgrößen, die als Zeitreihen zur Verfügung stehen. Jede Schicht beinhaltet einen Schwellvektor  $b$  und eine Wichtungsmatrix. Es gibt Wichtungsmatrizen für die Eingangsgrößen (*Input Weights*,  $IW$ ) und für die von der jeweiligen Schicht berechneten Ausgangsgrößen (*Layer Weights*,  $LW$ ). Die Dimensionen hängen von den Neuronenanzahlen  $S^1$  (Schicht 1) und  $S^2$  (Schicht 2) ab. Die Ein- und Ausgangsgrößen werden entsprechend gewichtet, mit den Schwellwerten addiert und zum Neuronen-Eingangsvektor  $n(t)$  der jeweiligen Schicht zusammengefasst. Für Schicht 1 folgt dementsprechend:

$$\text{Schicht 1, Eingang:} \quad n^1(t) = IW^{1,1} \cdot u(t) + LW^{1,2} \cdot y(t) + b^1 \quad (2.4)$$

$$\text{Schicht 1, Ausgang:} \quad a^1(t) = f(n^1(t)) \quad (2.5)$$

Mit dem Eingangsvektor  $n(t)$  und einer definierten Aktivierungsfunktion wird der Ausgangsvektor  $a(t)$  der jeweiligen Schicht berechnet. Die Aktivierungsfunktion hat entscheidenden Einfluss auf die Charakteristik des Netzes. Die verfügbaren Funktionen und deren Einflüsse auf das Netzverhalten sind Bestandteil der Untersuchungen in Kapitel 3. Der Block TDL (*Tapped Delay Lines*) fungiert als Speicher nach dem Prinzip "First In First Out". Wird der TDL-Wert auf  $d = 5$  gesetzt, zieht das Netz die Werte der vergangenen fünf Messzeitpunkte heran. Weiterhin muss festgelegt werden, ob das Netz seine eigenen berechneten Ausgangswerte (closed loop) für die Berechnung weiterer zukünftiger Werte heranzieht, oder ob es originale Messdaten (open loop) der Ausgangsgrößen verwenden soll (Abbildung 2.2). KNN werden standardmäßig open loop trainiert. Der Grund liegt darin, dass speziell beim Training das Prozessverhalten bestmöglich nachgebildet werden soll.

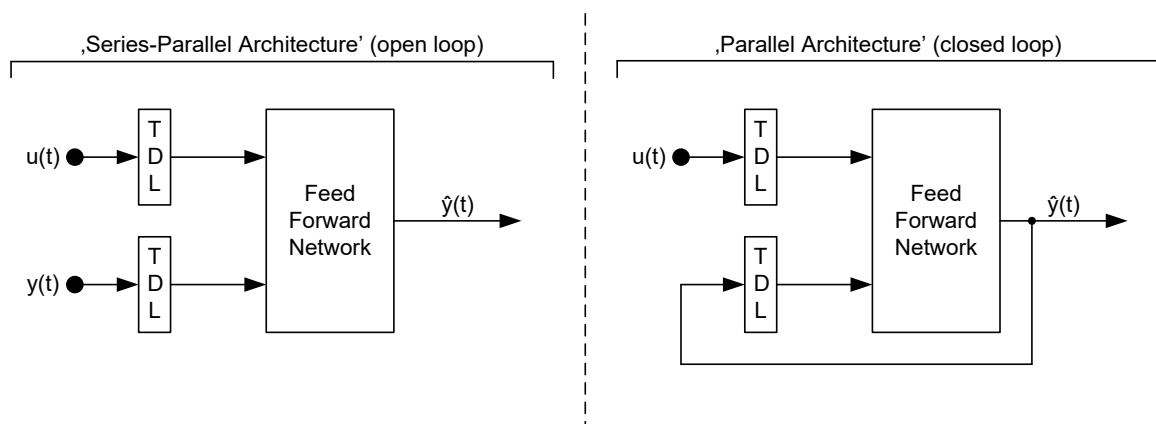


Abbildung 2.2: Grundlegende Architekturen für dynamische Netze (Basis: [67])



Gleichzeitig bedeutet die Verwendung der "open loop"-Architektur, dass das Netz nur Ausgangswerte zum unmittelbar nächsten Zeitpunkt berechnen kann (One Step Prediction, OSP). Eine Multi Step Prediction (MSP) ist nur im "closed loop"-Modus möglich.

Um den Datenfluss und den Rechenaufwand zu veranschaulichen, zeigt Abbildung 2.3 einen ausführlichen Signalfussplan eines zweischichtigen NARX-Netzes. Die Ausgangswerte der fünf Neuronen in der verborgenen Schicht werden nach den Gleichungen (2.4) und (2.5) berechnet. In der Ausgangsschicht entspricht die Anzahl der Neuronen der Anzahl der Ausgangsgrößen. Die Grafik zeigt ein vergleichsweise kleines Netz. *MATLAB* verwendet standardmäßig zehn verborgene Neuronen. Im Rahmen dieser Arbeit wurden Netze mit mehr als 100 Neuronen und mehreren Ein- und Ausgangsgrößen konfiguriert.

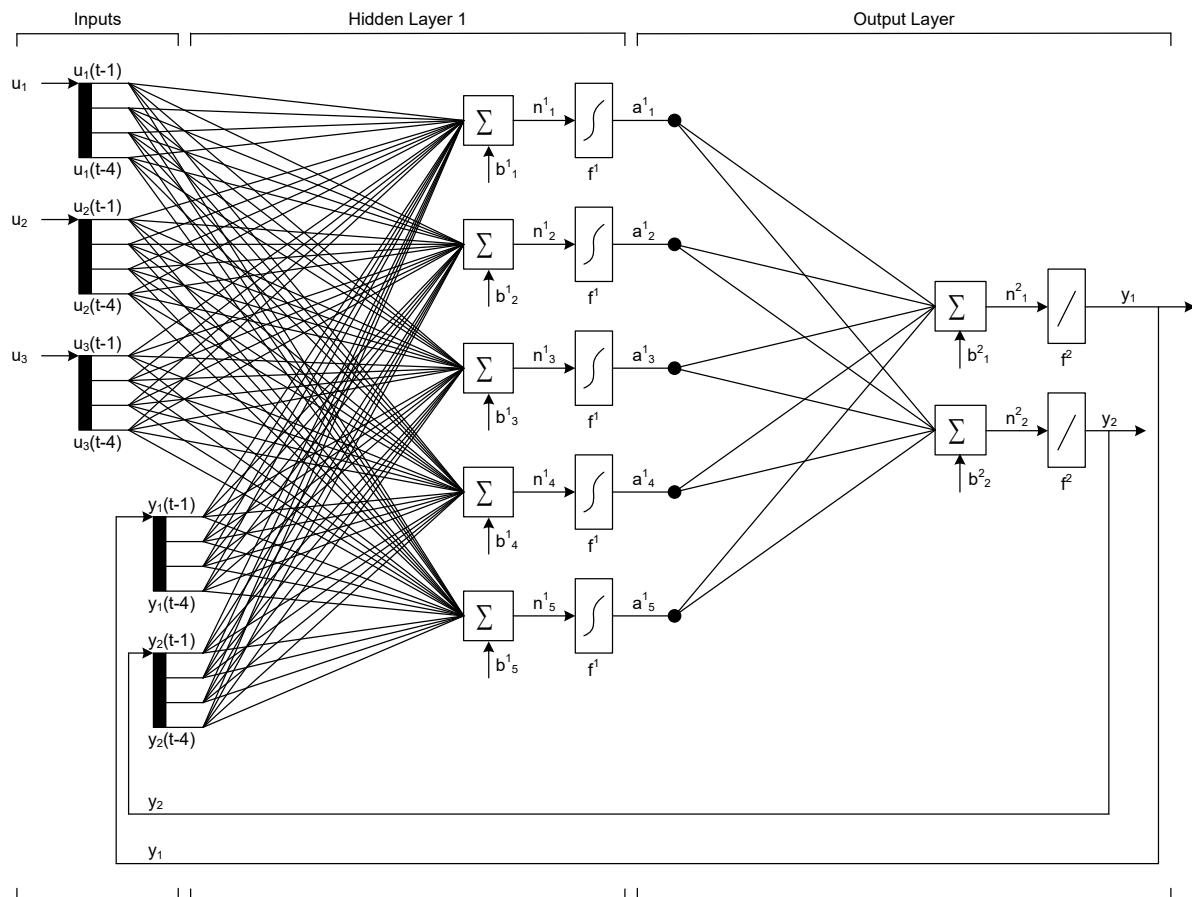


Abbildung 2.3: Signalfussplan eines zweischichtigen NARX-Netzes mit drei Eingangsgrößen, fünf verborgenen Neuronen, vier TDL, zwei Ausgangsgrößen und "closed loop"-Architektur

### 2.1.3 Konfigurationsmöglichkeiten

Im Rahmen dieser Arbeit wird auf den Einfluss folgender Entwurfparameter eingegangen:

- Anzahl und Auswahl der Ein- und Ausgangsgrößen,
- Auswahl der Trainingsmethode,
- Auswahl der Aktivierungsfunktion,
- Anzahl der verborgenen Neuronen und
- Anzahl der zurückliegenden Werte (TDL).

Wie in Abbildung 2.3 zu erkennen ist, erhöht sich die Anzahl notwendiger Rechenoperationen mit der Anzahl der Ein- und Ausgangsgrößen, der verborgenen Neuronen und der Höhe des TDL-Wertes. Diese Parameter bestimmen wesentlich den Rechenaufwand. Der Fokus der Arbeit liegt weniger auf der Minimierung des Rechenaufwands, sondern vielmehr auf der Identifikation einer geeigneten Netzkonfiguration.

Zu den internen Konfigurationsmöglichkeiten gehören die Auswahl geeigneter Trainingsmethoden und Aktivierungsfunktionen. Das Training des neuronalen Netzes ist elementar. Der Trainingsprozess beinhaltet das wiederholte Anpassen der Gewichte und der Schwellwerte, um eine vorgegebene Gütefunktion  $G$  zu minimieren. Die standardmäßig verwendete Gütefunktion für Feedforward-Netze in *MATLAB* berechnet die mittlere quadratische Abweichung (Mean Squared Error, MSE), die den durchschnittlichen quadratischen Fehler zwischen dem berechneten Netzausgang  $y_i$  und dem gemessenen Ausgang  $y_{mess}$  aller Ausgangsgrößen  $1 \leq i \leq N$  im Trainingszeitraum angibt.

$$G = MSE = \frac{1}{N} \sum_{i=1}^N (e_i)^2 = \frac{1}{N} \sum_{i=1}^N (y_{mess} - y_i)^2 \quad (2.6)$$

Eine geringe quadratische Abweichung bedeutet, dass sich die Netzfunktion in der Nähe der realen Übertragungsfunktion befindet. *MATLAB* bietet zwölf verschiedene Trainingsalgorithmen an, um die MSE-Funktion zu minimieren. Die meisten Optimierungsfunktionen basieren auf der Gradientenberechnung der Gewichte oder auf der Jacobi-Determinante der Netzfehler. Weiterhin wird bei allen Algorithmen das bekannte Verfahren der Backpropagation angewandt. Nach einem berechneten Fehler werden die Gewichte nicht in der ersten Schicht, sondern rückwärts beginnend von der Ausgabeschicht verändert. Der einfachste Optimierungsalgorithmus ist das Gradientenabstiegsverfahren. Das schrittweise Fort-

schreiten in Richtung des steilsten Abstiegs wird solange durchgeführt, bis keine numerische Verbesserung mehr auftritt und ein lokales Minimum vorliegt. Sind keine Abbruchkriterien erfüllt, werden die Iterationsschritte bis zur Konvergenz des Netzes durchgeführt. Laut [67] ist die Levenberg-Marquardt-Funktion (LM) die schnellste Optimierungsmethode und wird deshalb standardmäßig in *MATLAB* verwendet. Auch in [55, 59, 60, 69] wurde dieser Algorithmus erfolgreich angewandt. Eine Auflistung aller verfügbaren Trainingsalgorithmen sowie weiterführende Informationen zum LM-Algorithmus finden sich im Anhang A.

Einen weiteren entscheidenden Einfluss auf die Charakteristik des Netzes haben die Aktivierungsfunktionen. Diese lassen sich in folgende Gruppen einteilen:

- Funktionen mit Zweipunktverhalten,
- lineare Aktivierungsfunktionen,
- sigmoide Aktivierungsfunktionen und
- Basisfunktionen.

Sigmoide Aktivierungsfunktionen besitzen gegenüber unbegrenzten linearen Funktionen den Vorteil, dass ab einem bestimmten Maximalwert die Erregung keine zusätzliche Wirkung mehr erzielt. Es tritt für sehr große oder sehr kleine Eingangswerte eine Sättigung ein. Ähnlich ist es bei den Basisfunktionen. Die weit von Null entfernten Eingangswerte bewirken gegen Null strebende Ausgangswerte. Die sprunghaften Zweipunktfunktionen sowie die linearen Aktivierungsfunktionen werden bevorzugt in einfachen Netztopologien zur Klassifizierung und Mustererkennung eingesetzt. Besonders für mehrschichtige Netze, die nichtlineares Verhalten nachbilden, ist die Differenzierbarkeit der Aktivierungsfunktion eine wichtige Voraussetzung. Daher kommen meist lineare und sigmoide Funktionen zum Einsatz. Eine Übersicht aller 15 in *MATLAB* verfügbaren Aktivierungsfunktionen sowie deren Kennlinien, mathematische Beschreibung und Verwendungsmöglichkeiten sind im Anhang B dargestellt. Der Einfluss verschiedener sigmoider Aktivierungsfunktionen zur Time Series Prediction wird an einem Beispiel in [69] beschrieben.

Neben den genannten Konfigurationsmöglichkeiten spielt die generelle Datenbereitstellung zur Identifikation eines geeigneten Prozessmodells eine wichtige Rolle. Welche Daten sollen in welchem Umfang zum Training und Test zur Verfügung gestellt werden? Diese Frage wird ebenfalls in Kapitel 3 beantwortet.

## 2.2 Datensatzanalyse

### 2.2.1 Struktur des Datensatzes

Beim vorliegenden Datensatz handelt es sich um einen Auszug realer Messdaten einer amerikanischen Flachglasanlage. Der Datensatz erstreckt sich über einen Zeitraum von 23 Stunden und 58 Minuten. Es liegen insgesamt 22 gemessene Größen vor. Für jede Messgröße sind im genannten Messzeitraum jeweils 2.248 Messwerte vorhanden. Die Messungen erfolgten für alle Größen gleichzeitig, jedoch in unterschiedlichen Intervallen. Es wurde in jeder Minute mindestens einmal, maximal dreimal gemessen.

Ein Großteil der Daten liegt im englisch-amerikanischen Einheitensystem vor. Für ein einheitliches Bezugssystem wurde der gesamte Datensatz in das in Europa bekannte SI-Einheitensystem überführt. Die Umrechnungen und eine Auflistung aller Prozessgrößen sowie ausgewählte Kennzahlen sind im Anhang C dargestellt. Abbildung 2.4 zeigt eine vereinfachte Darstellung der Schmelz- und Abstehwanne und gibt eine Übersicht zur Anordnung der verfügbaren Prozessgrößen. Die Bedeutungen der Abkürzungen sowie weitere Informationen der einzelnen Prozessgrößen sind in Tabelle 2.1 aufgelistet.

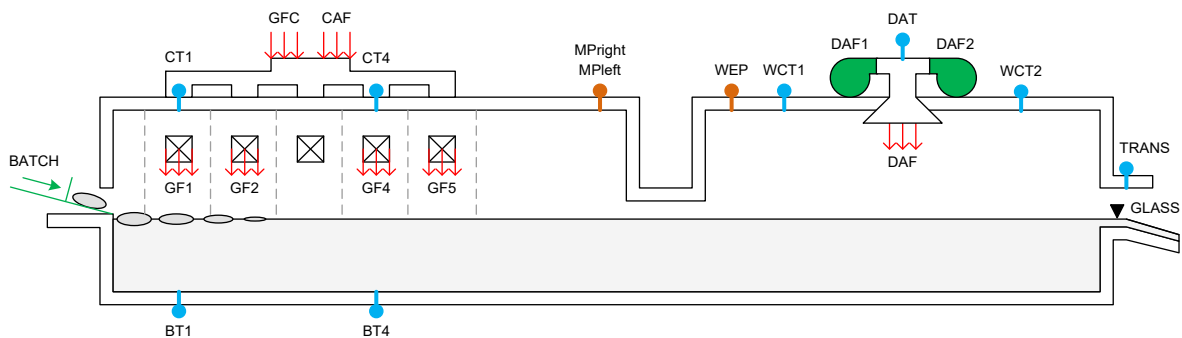


Abbildung 2.4: Vereinfachte Übersicht der Messanordnung

Tabelle 2.1: Auflistung der verfügbaren Messgrößen

Phys. Größe	Name	Bedeutung und Position	Einheit	Minimum	Maximum
Temperatur	CT1	Gewölbetemperatur in Brennerzone 1	K	1.580,05	1.598,15
	BT1	Bodentemperatur in Brennerzone 1	K	1.482,15	1.486,45
	CT4	Gewölbetemperatur in Brennerzone 4	K	1.757,85	1.761,95
	BT4	Bodentemperatur in Brennerzone 4	K	1.536,85	1.541,15
	WCT1	Gewölbetemperatur im vorderen Abstehbereich	K	1.424,05	1.427,35
	WCT2	Gewölbetemperatur im hinteren Abstehbereich	K	1.227,45	1.231,75
	TRANS	Gewölbetemperatur am Überlauf (Übergang zum Zinnbad)	K	1.370,80	1.373,55
	DAT	Temperatur der zugeführten Kühlluft	K	310,45	316,35
Volumenstrom	GFC	Zugeführte Gesamtgasmenge	Nm³/h	4.776,80	23.854,76
	GF1	Zugeführte Einzelgasmenge in Brennerzone 1	Nm³/h	726,58	4.983,40
	GF2	Zugeführte Einzelgasmenge in Brennerzone 2	Nm³/h	841,19	5.013,75
	GF4	Zugeführte Einzelgasmenge in Brennerzone 4	Nm³/h	1.027,22	5.075,18
	GF5	Zugeführte Einzelgasmenge in Brennerzone 5	Nm³/h	1.033,51	4.158,65
	CAF	Zugeführte Verbrennungsluftmenge	Nm³/h	227.485,19	281.903,14
	DAF	Zugeführte Kühlluftmenge (Abstehwanne)	Nm³/h	469,56	573,97
Druck	WEP	Herdraumdruck im vorderen Abstehbereich	Pa	9,25	19,95
	MPleft	Linker Herdraumdruck im hinteren Schmelzbereich	Pa	-13,88	14,94
	MPright	Rechter Herdraumdruck im hinteren Schmelzbereich	Pa	-17,24	13,12
Geschwindigkeit	DAF1	Prozentuale Geschwindigkeit der Kühlluftzufuhr von Lüfter 1	%	0,00	84,01
	DAF2	Prozentuale Geschwindigkeit der Kühlluftzufuhr von Lüfter 2	%	0,00	64,98
	BATCH	Prozentuale Geschwindigkeit der Gemengezufuhr	%	35,72	39,39
Glasstand	GLASS	Glasstand der Wanne im hinteren Abstehbereich	mm	-0,30	0,22

Beim vorliegenden Datensatz muss berücksichtigt werden, dass die Daten im geregelten Betrieb der Anlage aufgenommen wurden. Die meisten Größen sind absolut, einige jedoch relativ gemessen (z. B. GLASS). Bei DAF1, DAF2 und BATCH handelt es sich um prozentuale Geschwindigkeiten. Besonders bei der Gemeengezufuhr BATCH ist diese Maßeinheit untypisch. Üblicherweise wird die Gemeengezufuhr in Tonnen pro Tag [t/d] angegeben. Zudem sind die Wertebereiche der Daten sehr unterschiedlich, weshalb eine Normierung unabdingbar ist. Weiterhin wurde zu einer besseren grafischen Darstellung der erste Messzeitpunkt auf 00:00:00 Uhr zurückgesetzt.

## 2.2.2 Interpretation der Messdaten

Im Folgenden werden ausgewählte Prozessgrößen näher betrachtet und deren Wechselwirkungen untereinander beschrieben.

### Schmelzbereich

Abbildung 2.5 stellt den zeitlichen Verlauf der Gemeengezufuhr dar. Die Gemeengezugabe ist zusammen mit den Gas- und Luftmengen eine der Stellgrößen im Schmelzbereich.

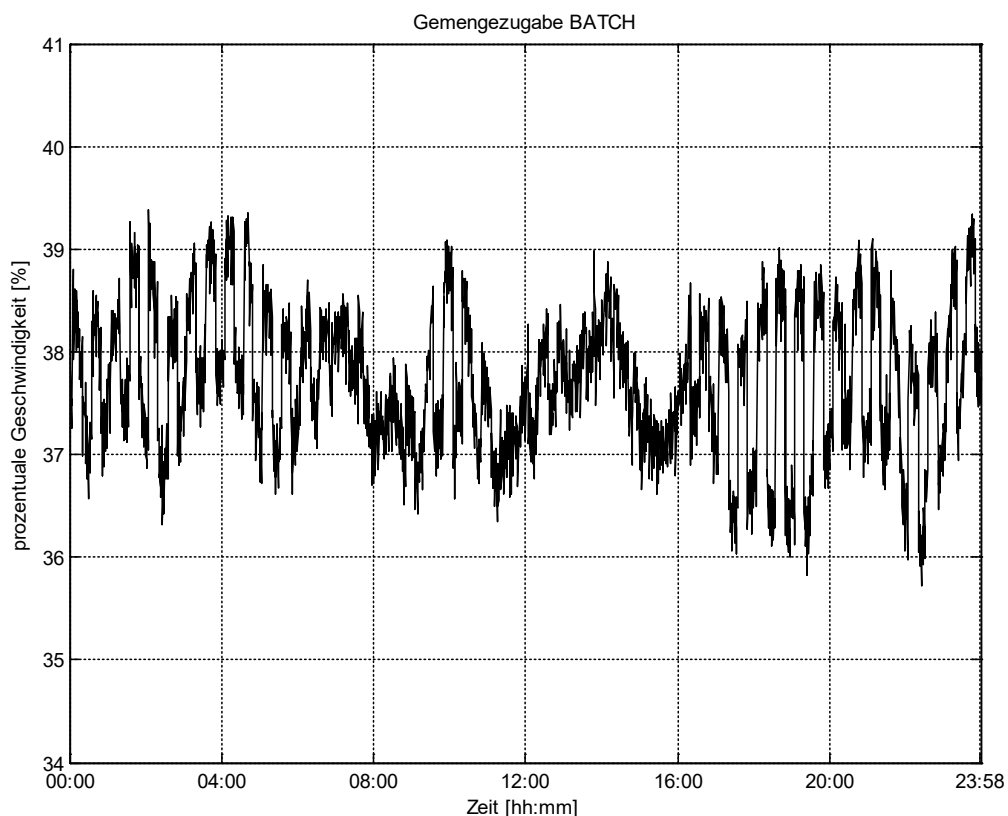


Abbildung 2.5: Zeitlicher Verlauf von BATCH im gesamten Messzeitraum

Das vergleichsweise kalte Gemenge wird am Ofeneingang auf das Glasbad geschoben und in den folgenden Brennerzonen vollständig geschmolzen. Die Geschwindigkeit der Gemengezugabe hat Einfluss auf die nachfolgenden chemischen Prozesse. Wird das Gemenge schneller hineingeschoben, ist der Bedeckungsgrad der Glasschmelze größer. Es bildet sich mehr Schaum, die Deckentemperaturen steigen aufgrund der Reflektion. Die Abbildung 2.5 zeigt, dass über den gesamten Zeitraum die prozentuale Geschwindigkeit zwischen 35 - 40 % liegt. Weiterhin sind in regelmäßigen Abständen sprunghafte Änderungen erkennbar. Der Ursprung dieser sprunghaften Änderungen wird erst nach der Gegenüberstellung mit der Gesamtgasmenge GFC und der zugeführten Verbrennungsluftmenge CAF deutlich (Abbildung 2.6). Die Grafik zeigt die zeitlichen Verläufe für die ersten vier Stunden. Alle drei Prozessgrößen zeigen zu den gleichen Zeitpunkten sprunghafte Änderungen. Diese Sprünge stellen die Feuerwechselzeitpunkte dar. Alle 15 Minuten findet ein Feuerwechsel statt. Es ist zudem erkennbar, dass die Sprünge der Gemengezugabe und der Verbrennungsluft entgegengesetzt sind.

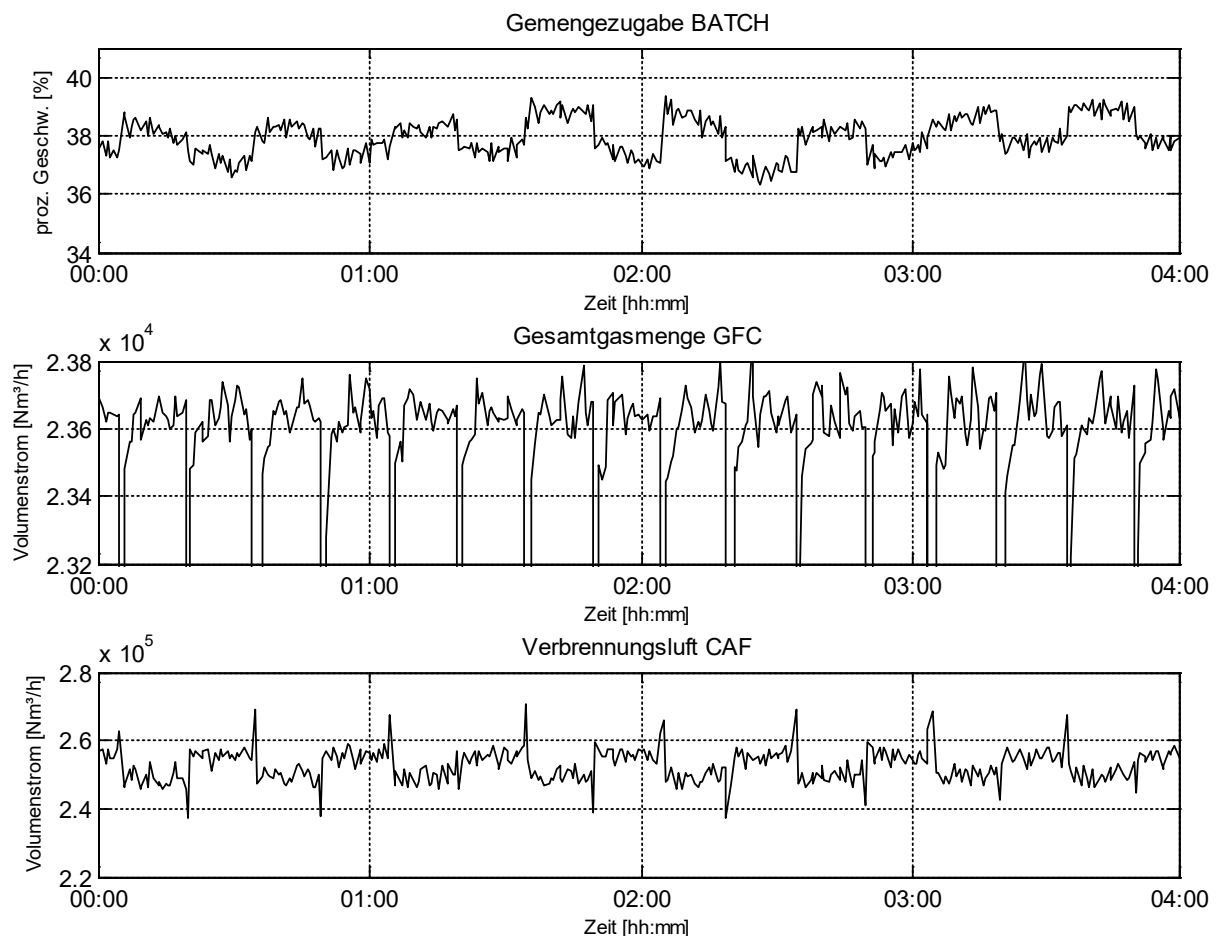


Abbildung 2.6: Sprunghafte Änderungen von BATCH, GFC und CAF

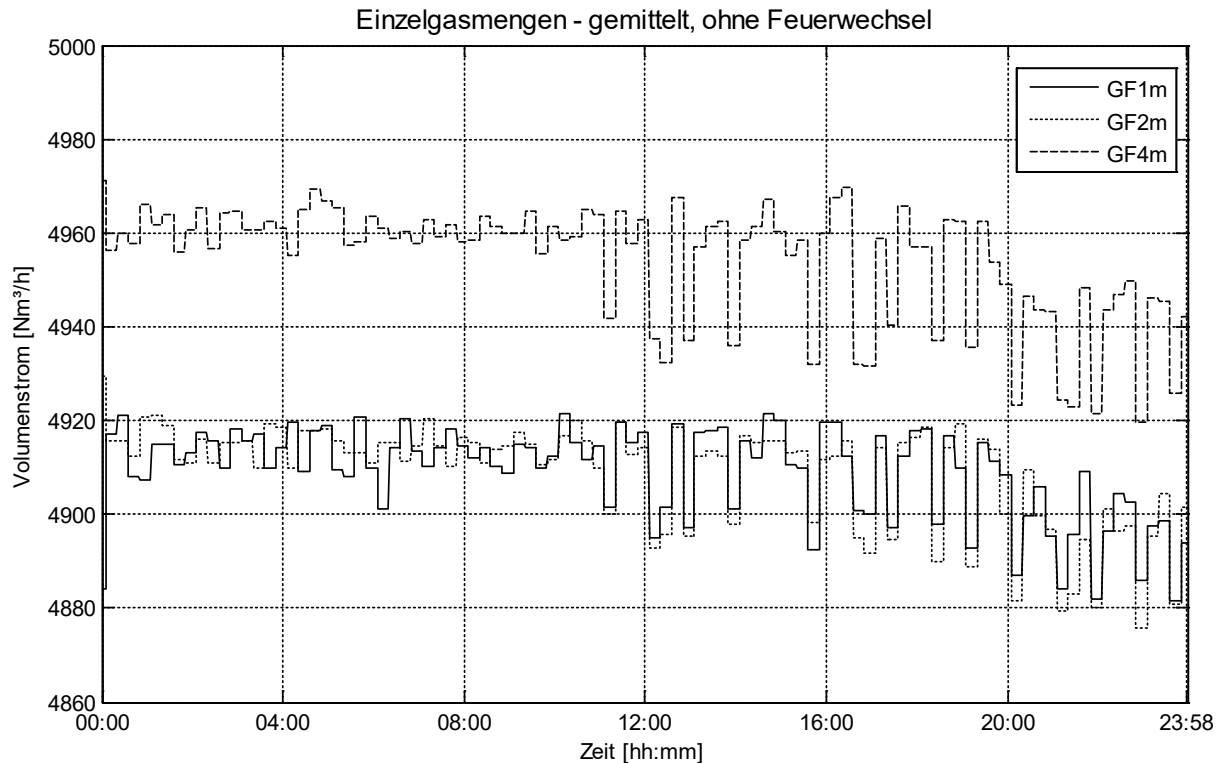


Abbildung 2.7: Gegenüberstellung der Einzelgasmengen GF1, GF2 und GF4

Abbildung 2.7 stellt die zeitlichen Verläufe der Einzelgasmengen GF1, GF2 und GF4 direkt gegenüber. Die Feuerwechselsprünge wurden entfernt und die Werte in den Feuerphasen gemittelt. Es ist deutlich zu erkennen, dass die Sprünge zur gleichen Zeit und in den meisten Fällen auch in die gleiche Richtung erfolgen. Somit kann geschlossen werden, dass die Gesamtgasmenge GFC zu unterschiedlichen Anteilen an alle Brenner verteilt wird. Wird die Gesamtgasmenge erhöht, erhöhen sich automatisch alle Einzelgasmengen. In den ersten beiden Zonen wird annähernd die gleiche Menge Gas verfeuert. In der vierten Zone wird die größte Menge Gas verbrannt, während in der letzten, fünften Brennerzone (nicht dargestellt) deutlich weniger Gas zugeführt wird. Nach ca. zwölf Stunden hat eine sichtbare Reduzierung der zugeführten Gasmengen stattgefunden. Die Stufenhöhe hat ab diesem Zeitpunkt deutlich zugenommen. Eine Reduzierung der Gasmenge resultiert in einer Reduzierung des Wärmeeintrags im Schmelzbereich. Abbildung 2.8 veranschaulicht die Auswirkungen der Gasmengenzufuhr auf die Temperaturen im Schmelzbereich.



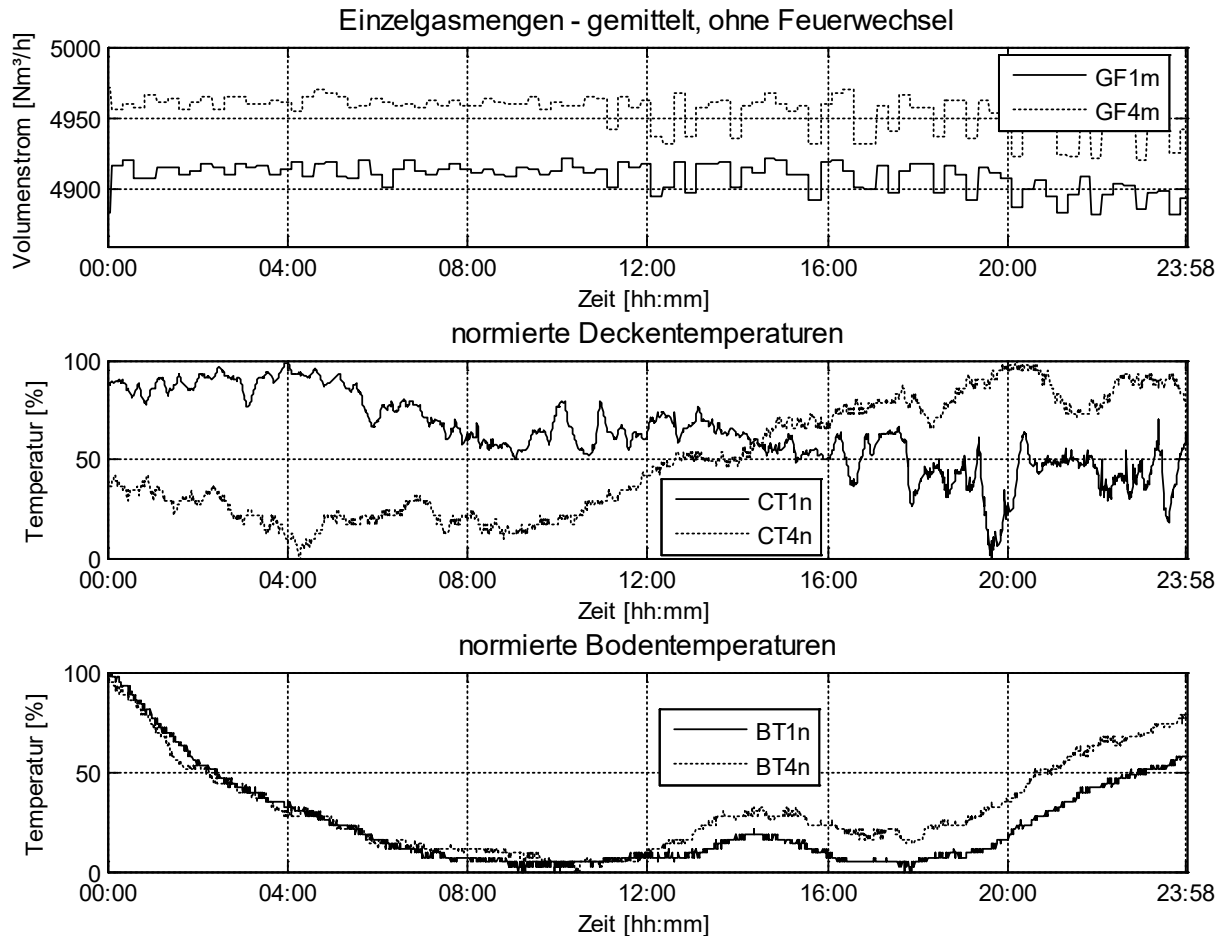


Abbildung 2.8: Gegenüberstellung der Gasmengen und Temperaturen im Schmelzbereich

Die Decken- und Bodentemperaturen sind aufgrund der unterschiedlichen Wertebereiche normiert, um die Wechselwirkungen besser zu veranschaulichen. Die Deckentemperatur CT1 in der ersten Zone verhält sich entsprechend der zugeführten Einzelgasmenge erwartungsgemäß. Die zunehmenden Gasänderungen ab 12:00 Uhr führen zu einer geringeren Deckentemperatur, welche ebenfalls größeren Schwankungen unterworfen ist. Einen gegenläufigen Trend zeigt jedoch die Deckentemperatur in Zone 4. Obwohl sämtliche Gasmengen mit zunehmender Zeit im Mittel reduziert wurden, steigt die Deckentemperatur in dieser Zone an. Möglicherweise wirken sich die anfangs größeren Gasmengen erst nach mehreren Stunden auf die Deckentemperatur in dieser Zone aus. Diese großen Zeitverzögerungen sind für turbulente Wirbelprozesse jedoch unrealistisch. Der Temperaturverlauf ist zudem im Vergleich zur ersten Zone deutlich ruhiger. Der Grund liegt darin, dass das isolierend wirkende Gemenge auf der Schmelzoberfläche in der Zone 4 weitestgehend geschmolzen ist und somit die Wärmerückstrahlung geringer ausfällt.

Die Bodentemperaturen beider Zonen zeigen einen nahezu identischen Verlauf. Auch hier ist der Trend den reduzierten Gasmengen entgegengesetzt. Der vergleichsweise glatte und ähnliche Verlauf beider Temperaturen kann auf die sehr langsamen Strömungsprozesse in der Glasschmelze zurückgeführt werden. Die Bodentemperaturen sind gegenüber den turbulenten Feuerungsprozessen im Oberofen unempfindlich. Ein direkter Zusammenhang zu den Gasmengen ist nicht erkennbar.

Die bisherigen Darstellungen führten teilweise zu unerwarteten Ergebnissen. Eventuell ist eine klassische Betrachtung des Ursache-Wirkung-Prinzips bei diesem komplexen, nicht-linearen Prozess anhand der vorliegenden Messdaten nicht zielführend. Es ist ebenso denkbar, dass die zugeführten Gasmengen kontrolliert reduziert wurden, weil die Temperaturen in Zone 4 gestiegen sind. Die Ursachen für den Temperaturanstieg können mitunter völlig anderer Natur und müssen nicht zwingend aus dem Datensatz ablesbar sein. Bezogen auf den Schmelzbereich sind die Ergebnisse der Datensatzanalyse der bisher dargestellten und aller weiteren Prozessgrößen noch einmal zusammengefasst.

- Alle 15 Minuten findet ein Feuerwechsel statt.
- Die Verteilung der Gesamtgasmenge GFC erfolgt in verschiedenen Verhältnissen gleichmäßig an alle Brenner.
- Die zugeführten Gasmengen weisen von Feuerperiode zu Feuerperiode teilweise große Schwankungen auf (bis zu 40 Nm<sup>3</sup>/h).
- Die Gemengezufuhr BATCH und die Verbrennungsluft CAF zeigen im entgegengesetzten Sinn sprunghafte Änderungen.
- Die Verbrennungsluft CAF wird in Längsrichtung konstant unsymmetrisch verteilt.
- Die Deckentemperatur CT1 verhält sich gegenüber der zugeführten Gasmenge GF1 annähernd erwartungsgemäß.
- Ein direkter Zusammenhang der Boden- und restlichen Deckentemperaturen zu den lokalen Gasmengen ist nicht erkennbar.
- Die Bodentemperaturen BT1 und BT4 zeigen einen annähernd gleichen Verlauf.
- Die Temperaturen CT4, BT1 und BT4 ändern sich im gleichen Umfang und in der gleichen Geschwindigkeit.
- Die Deckentemperaturen (ca. 1.650 K) sind deutlich höher als die Bodentemperaturen (ca. 1.500 K).

### Abstehbereich

Im Abstehbereich stehen vier weitere Temperaturmessstellen zur Verfügung. In dieser Zone ist der Schmelzprozess vollständig abgeschlossen und die Glasschmelze homogenisiert. Die Hauptaufgabe ist die kontrollierte Kühlung auf ein vorgegebenes Temperaturniveau. Als Stellgrößen stehen die zugeführte Kühlluftmenge DAF und die prozentualen Geschwindigkeiten der Lüfter zur Verfügung. Eine Analyse der Daten hat ergeben, dass die beiden abwechselnd eingeschalteten Lüfter DAF1 und DAF2 zusammen den exakt gleichen Verlauf wie die zugeführte Kühlluftmenge DAF aufweisen. Eine Erhöhung der Kühlluftmenge erfordert eine proportionale Erhöhung der Lüftergeschwindigkeit. Aus diesem Grund wird vornehmlich die Kühlluftmenge DAF als einzige Stellgröße im Abstehbereich betrachtet. In der Abbildung 2.9 sind die zeitlichen Verläufe der Kühlluftmenge und der normierten Temperaturen dargestellt.

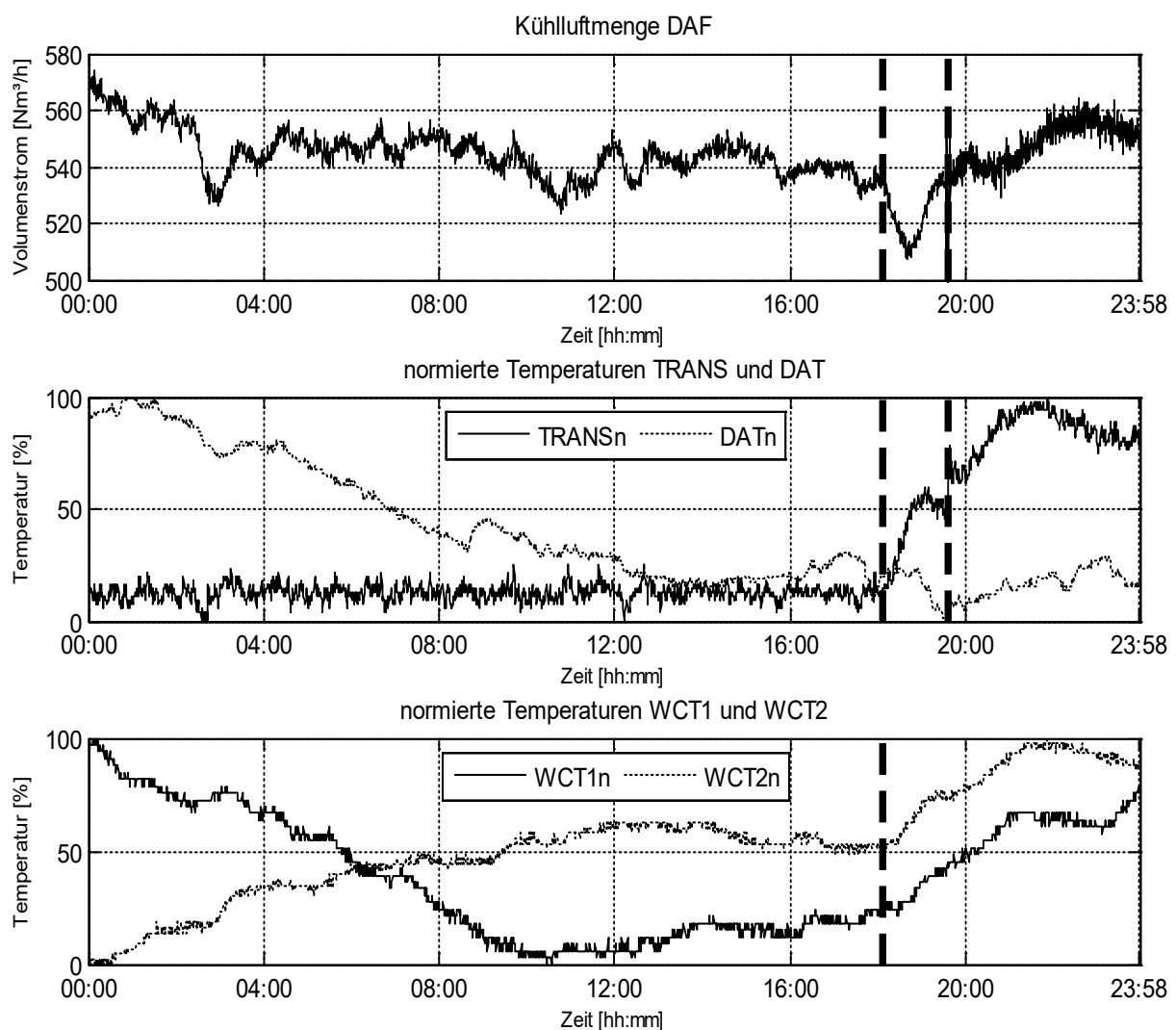


Abbildung 2.9: Zeitliche Verläufe der Prozessgrößen in der Abstehwanne

Nach Abbildung 2.4 ist die Temperaturmessstelle WCT1 vor und die Temperaturmessstelle WCT2 nach der Kühlluftzufuhr angeordnet. Die Temperatur TRANS wird direkt am Ende der Abstehtwanne gemessen, bevor die Glasschmelze über den Lippenstein auf das Zinnbad übergeht. Der zeitliche Verlauf der zugeführten Kühlluftmenge zeigt zwei markante Punkte, die durch Linien in den Diagrammen gekennzeichnet sind.

Der erste markante Zeitpunkt ist nach ca. 18 Stunden. Innerhalb von 35 Minuten wurde die Kühlluftmenge um ca.  $26 \text{ Nm}^3/\text{h}$  reduziert. Die Ausgangstemperatur TRANS steigt ab diesem Zeitpunkt an und pegelt sich nach über fünf Stunden bei ca. 80 % ihres Temperaturwertebereichs ein, was einer Änderung von  $1,8 \text{ K}$  entspricht. Der Verlauf ähnelt einer klassischen Sprungantwort eines proportionalen Verzögerungsgliedes mindestens zweiter Ordnung. In den ersten 18 Stunden weist die Temperatur TRANS einen konstanten, scheinbar geregelten Verlauf auf. Eventuell erfolgte zu diesem Zeitpunkt ein Sollwertsprung. Der weitere Verlauf der Kühlluftmenge widerspricht aber dem Verlauf von TRANS. Die Kühlluft wird wieder erhöht, doch ein Einfluss auf die Ausgangstemperatur ist nicht erkennbar. Beim Betrachten der Kühllufttemperatur DAT fällt weiterhin auf, dass die Temperatur über den gesamten Messzeitraum sinkt. Eine erhöhte und zudem kühlere Kühlluftzufuhr müsste zu einer Temperaturabnahme der nachstehenden Deckentemperaturen führen. Dies ist bei der Ausgangstemperatur TRANS und der Deckentemperatur WCT2 jedoch nicht der Fall.

Bei der zweiten Markierung (19:30 Uhr) erfolgte ein Lüfterwechsel. Dieser Wechsel brachte Unruhe in Form von Rauschen in den Kurvenverlauf der Kühlluftmenge DAF. Der weitere Kurvenverlauf liegt nicht vor, so dass keine Aussage getroffen werden kann, ob dieses Rauschen wieder abklingt. Die durch den Lüfterwechsel verursachte Unruhe ist auch in der Ausgangstemperatur TRANS erkennbar. Hier setzte ein regelrechter Temperatursprung ein.

Ein Vergleich der zeitlichen Verläufe von WCT1 und DAF lässt die Vermutung zu, dass die Kühlluft nach der Deckentemperatur WCT1 gesteuert wird. Zu Beginn der Messungen erreicht die Deckentemperatur ihren Maximalwert. Diese hohe Temperatur verursacht ebenfalls eine maximale Kühlung. Mit der Zeit nimmt die Deckentemperatur ab. Auch bei der Kühlluftmenge ist eine Reduzierung zu verzeichnen. Bis ca. 18:00 Uhr ist dieser Zusammenhang umgekehrt auch für die Deckentemperatur WCT2 gültig. Eine Zunahme der Kühlluftmenge führt zu einer Temperaturabnahme von WCT2. Anschließend gelten diese Wechselwirkungen nicht mehr. Es bleibt auch hier festzuhalten, dass allein aus der Betrachtung der zeitlichen Verläufe keine direkten oder teilweise nur bedingten Zusammenhänge nach dem Ursache-Wirkungs-Prinzip erkennbar sind. Die Temperaturen WCT1 und

WCT2 sind anscheinend unregelmäßige, absolute Messgrößen, während die Ausgangstemperatur TRANS offenbar als Regelgröße vorliegt. In den ersten 18 Stunden schwankt die Temperatur TRANS mit  $\pm 0,25$  K um den Wert 1.371,15 K. Aus regelungstechnischer Sicht ist die Prozessidentifikation im laufenden Betrieb möglich, jedoch müssen einige Aspekte zusätzlich beachtet werden. Verrauschte Signale sind nur bedingt nutzbar. Ausreißer und Messrauschen verfälschen die Ergebnisse stark. Eine weitere Notwendigkeit besteht in der Eliminierung von Trend- und Drifterscheinungen. Diese sind typisch für längere industrielle Messdatensätze.

Abschließend werden die Prozessgrößen WEP und GLASS näher betrachtet (Abbildung 2.10). Ein konstanter Herdraumdruck WEP ist Voraussetzung für einen optimalen Schmelzprozess. Eine Messung des Druckes bietet die Möglichkeit, Falschlufteinbrüche zu erkennen, welche den Herstellungsprozess negativ beeinflussen. Das Messsignal ist jedoch sehr verrauscht und zeigt viele Ausreißer. Für eine Prozessidentifikation ist diese Signalqualität unzureichend.

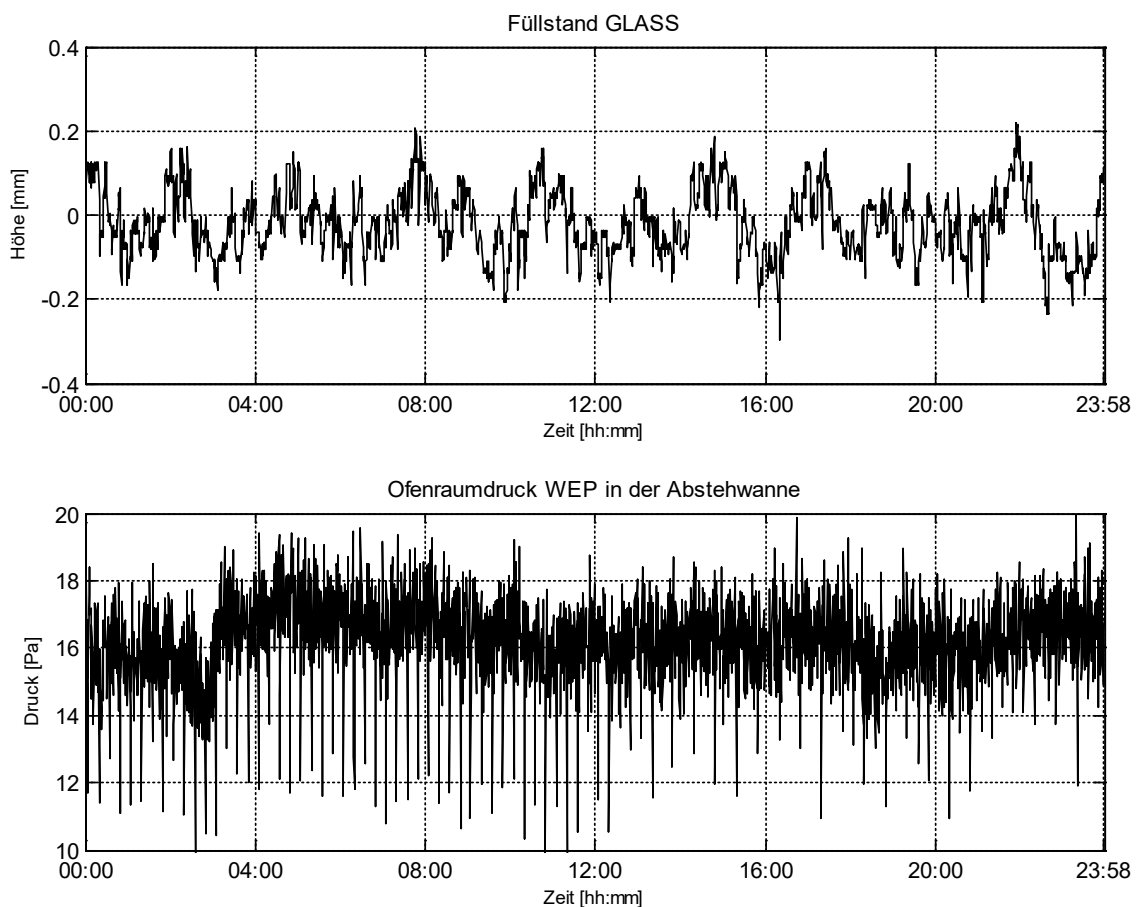


Abbildung 2.10: Zeitliche Verläufe von GLASS und WEP

Auch beim Glasstand GLASS am Ende des Abstehbereichs müssen Einschränkungen in der Nutzbarkeit des Signals hingenommen werden. Bei dieser Messgröße handelt es sich um eine relativ gemessene Regelgröße, die über den gesamten Zeitraum annähernd gleichmäßig schwankt. Der Sollwert liegt vermutlich bei 0 mm und die Abweichungen liegen zumeist in einer Spanne von  $\pm 0,2$  mm. Der Glasstand ist ebenfalls qualitätsbestimmend für das Endprodukt. Ein zu hoher Füllstand bedeutet eine erhöhte Glasschmelzenzufuhr auf das nachfolgende Zinnbad, wodurch vermehrte Stelleingriffe zur gleichmäßigen Ausbreitung eines Glasbandes erforderlich sind. Eine Abhängigkeit beider Größen zu anderen Prozessgrößen im Abstehbereich ist nicht erkennbar. Die wichtigsten Erkenntnisse für den Abstehbereich sind nachfolgend noch einmal zusammengefasst.

- Es besteht kein direkter Zusammenhang zwischen den beiden Deckentemperaturen WCT1 und WCT2.
- Es kann von der Bodentemperatur BT4 auf die Deckentemperatur WCT1 geschlossen werden. Beide Verläufe gleichen sich. Die Zeitverzögerung beträgt ca. zwei Stunden.
- Die qualitätsbestimmende Deckentemperatur TRANS durchläuft nach ca. 18 Stunden einen Sollwertsprung. Ob die Kühlluftmenge DAF dabei als Stellgröße fungiert, kann nicht abschließend geschlussfolgert werden.
- Die Kühlluftmenge DAF wird nach der Deckentemperatur WCT1 gesteuert. Je höher die Deckentemperatur, desto höher die zugeführte Kühlluftmenge.
- Das Aufheiz- und Abkühlverhalten der Ausgangstemperatur TRANS ist bei Kühlluftänderungen sehr unterschiedlich und kann nicht eindeutig identifiziert werden.
- Die steuerbare Kühlluftzufuhr DAF ist die einzige Möglichkeit Stelleingriffe in der Abstehwanne vorzunehmen.
- Alle Signale der Herdraumdrücke (MPleft, MPright, WEP) sind stark verrauscht und werden nicht weiter für die Prozessidentifikation berücksichtigt.
- Der geregelte Füllstand GLASS schwankt gleichmäßig um den Sollwert 0 mm.
- Eine Abhängigkeit des Glasstandes von der Mengeneingabe BATCH ist aus der Betrachtung der zeitlichen Verläufe nicht erkennbar.

Die dargestellten zeitlichen Verläufe geben einen ersten Eindruck über den zu regelnden Prozess. Um eine bessere grafische Darstellung zu ermöglichen wurden viele Verläufe auf den jeweiligen Wertebereich normiert. Eine ausführliche Tabelle aller verfügbaren Prozessgrößen mit statistischen Kennzahlen ist im Anhang C enthalten. Im Anhang D sind alle zeitlichen Verläufe noch einmal übersichtlich dargestellt.

An dieser Stelle muss festgehalten werden, dass es sich um einen reduzierten Datensatz handelt. An einer Flachglasanlage werden im Schmelz- und Abstehbereich weitaus mehr Temperaturen gemessen. Die vorhandenen Messdaten zeigen untereinander kaum Zusammenhänge. Eine umfassende Analyse des Datensatzes mit von *MATLAB* bereitgestellten Identifikationstechniken (*System Identification Toolbox*) wurde ebenfalls durchgeführt und brachte keine nutzbaren Erkenntnisse. Es wurden Übertragungsfunktionen verschiedener Ordnungen, Zustandsmodelle und nichtlineare Modelle erstellt. Die Verifizierung mit weiteren Testdaten war nicht zufriedenstellend.

### 2.2.3 Aufbereitung des Datensatzes

Der Datensatz enthält Messungen in unterschiedlichen Intervallen. Für die programmier-technische Realisierung mit *MATLAB* werden gleichmäßige Abstände zwischen den Messzeitpunkten vorausgesetzt. Infolgedessen wurde der Datensatz bis in den Sekundenbereich erweitert. Abbildung 2.11 zeigt die einzelnen Schritte zur Vergrößerung des Datensatzes am Beispiel der Kühlluftmenge DAF. In der ersten Spalte sind die ersten drei originalen Messzeitpunkte und deren Werte aufgelistet. Für die Identifikation des Prozessverhaltens spielen Datum und Uhrzeit keine Rolle. Der erste Messzeitpunkt wurde auf Null zurückgesetzt. Ausgehend von diesem Zeitpunkt wurden die Abstände zwischen allen weiteren Zeitpunkten berechnet und kumuliert (Spalte 2), bevor in Spalte 3 der Datensatz zu jeder Sekunde erweitert wurde. Dadurch ist die Anzahl der Messzeitpunkte signifikant auf 86.281 Werte pro Messgröße gestiegen. Diese Menge an Daten ist für die nachfolgenden Untersuchungen deutlich zu groß und überschreitet die verfügbare Rechenkapazität. Bevor eine notwendige Reduzierung des Datensatzes erfolgt, ist es wichtig zu klären, mit welchen Messwerten die neu hinzugefügten Messzeitpunkte gefüllt werden. Die leeren Zellen können u. a. mit dem jeweils letzten bekannten Wert aufgefüllt werden. Das bedeutet gleichzeitig, dass sich sprunghafte Änderungen bei physikalischen Größen ergeben, die sich nicht sprunghaft ändern können (z. B. Temperaturen). Aus diesem Grund wurde zwischen den Messwerten linear interpoliert.

Originaler Messzeitpunkt mit Messwert			Zurückgesetzter Messzeitpunkt		Pro Sekunde hinzu- gefügte Messzeitpunkte	
27.03.2007 20:26:14	569,09		00:00:00	569,09	00:00:00	569,09
27.03.2007 20:27:14	570,27		00:01:00	570,27	00:00:01	
27.03.2007 20:27:54	568,52		00:01:40	568,52	00:00:02	
•			•		00:00:03	
•			•		00:00:04	
•					•	
					•	
					00:00:59	
					00:01:00	570,27
					00:01:01	
					00:01:02	
					00:01:03	
					00:01:04	
					•	
					•	
					00:01:39	
					00:01:40	568,52
					00:01:41	
					00:01:42	
					00:01:43	
					00:01:44	
					•	
					•	
					•	

Abbildung 2.11: Aufbereitung und Erweiterung der Messzeitpunkte

Die lineare Interpolation und anschließende Reduzierung des Datensatzes führt zu einem nicht zu vermeidenden Nebeneffekt, der durch die Abbildungen 2.12 und 2.13 veranschaulicht wird. Abbildung 2.12 zeigt für die Prozessgröße BATCH die originalen Messwerte für die ersten 600 Sekunden. Die unterschiedlichen Messintervalle sind erkennbar. Zusätzlich sind beide Möglichkeiten zur Vervollständigung des Datensatzes dargestellt, wobei die linear interpolierte Kurve näher betrachtet werden soll. Die Reduzierung des Datensatzes auf konstante Intervalle mit der Länge von 30 Sekunden ist in Abbildung 2.13 durch die rechteckigen Markierungen veranschaulicht. Mit diesen Daten kann das Netz trainiert und getestet werden. Eine lineare Verbindung dieser "neuen" Messzeitpunkte zeigt, dass einige originale Messdaten nicht mehr in die Berechnungen mit eingehen. Dieser Informationsverlust lässt sich aufgrund der Notwendigkeit einer konstanten Schrittweite nicht vermeiden.



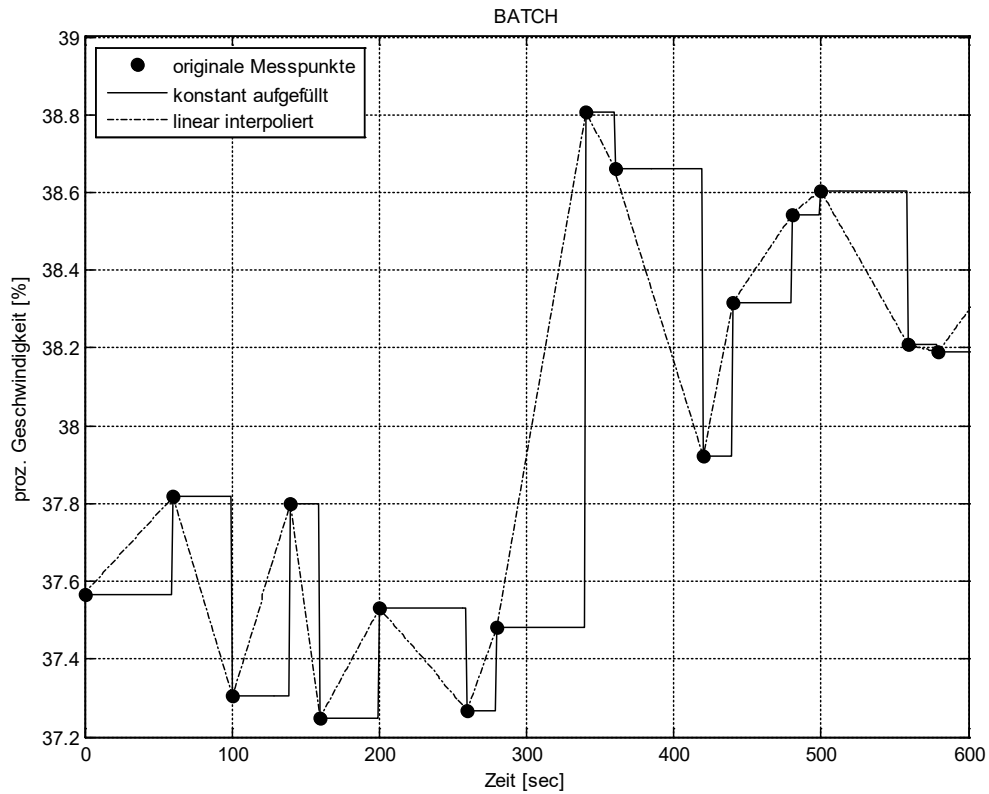


Abbildung 2.12: Lineares und konstantes Auffüllen des Datensatzes

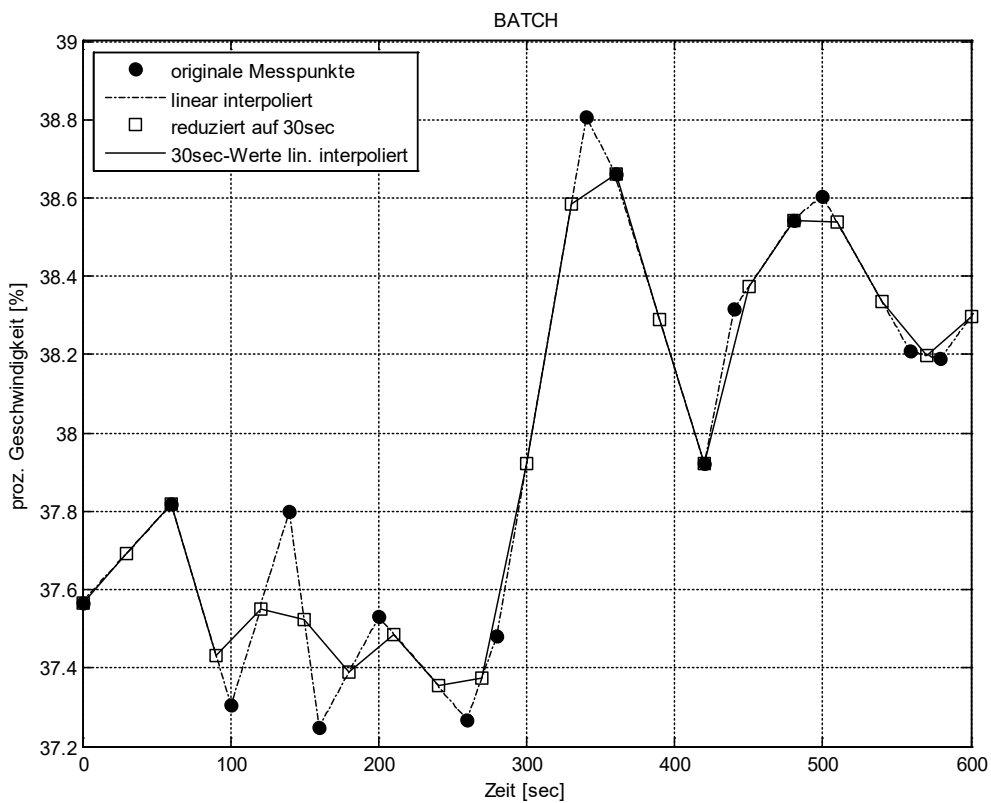


Abbildung 2.13: Informationsverlust durch gleichmäßige Verteilung der Messzeitpunkte

Eine Reduzierung des Datensatzes ist in wählbaren Intervallen möglich. Die meisten Messungen erfolgten nach 20 oder 40 Sekunden. Bei einer Schrittweite von 20 Sekunden ergeben sich insgesamt 4.315 Messzeitpunkte. Um im Rahmen der ursprünglichen Datenmenge zu bleiben, wurde eine Schrittweite von 30 Sekunden gewählt. Dies entspricht 2.877 Messzeitpunkten und einer Gesamtdatenmenge von 66.171 Werten.

Zusätzlich wurde ein weiterer Datensatz aufgestellt, bei dem die originalen Messdaten stark vereinfacht wurden. Die originalen Messreihen sind teilweise verrauscht oder unterliegen größeren Schwankungen. In den meisten Fällen sind aber eindeutige Trends erkennbar. Die Daten wurden durch Mittelwertbildung oder Interpolation so stark vereinfacht, dass lediglich deren grober Trend dargestellt wird. Abbildung 2.14 zeigt Ausschnitte der originalen und stark vereinfachten Verläufe für vier exemplarische Prozessgrößen. Dieser vereinfachte Datensatz soll zusätzlich zur Identifikation geeigneter neuronaler Netze herangezogen werden.

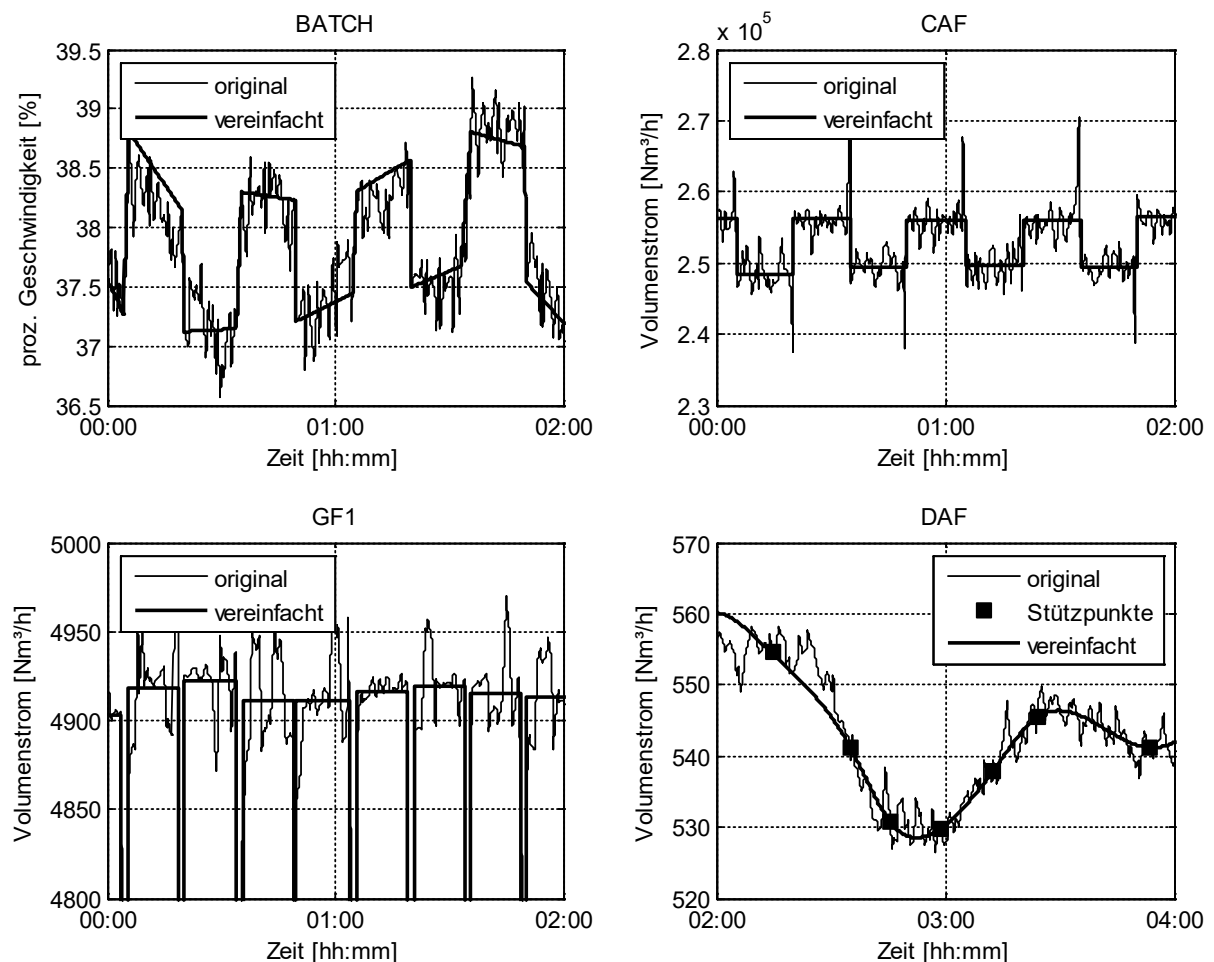


Abbildung 2.14: Vergleich von originalen und vereinfachten Messdaten

## 2.3 Konfigurationsassistent und Bewertungskriterien

Die *Neural Network Toolbox* enthält umfangreiche Funktionen zum Entwurf von Künstlichen Neuronalen Netzen. Ein NARX-Netz kann in wenigen Schritten erstellt werden. Der wesentliche Aufwand besteht in der Konfiguration der Netze. Wie viele und welche Eingangsgrößen sollen verwendet werden? Welche Ausgangsgrößen soll das Netz berechnen? Wie soll die Struktur (u. a. Neuronenanzahl, TDL) des Netzes aussehen? Welche Daten sollen dabei verwendet werden? Vor jeder Simulation sind diese Änderungen im Quellcode vorzunehmen. Aus diesem Grund wurde eine grafische Benutzeroberfläche programmiert, die die Konfiguration der Netze erleichtert (Abbildung 2.15).

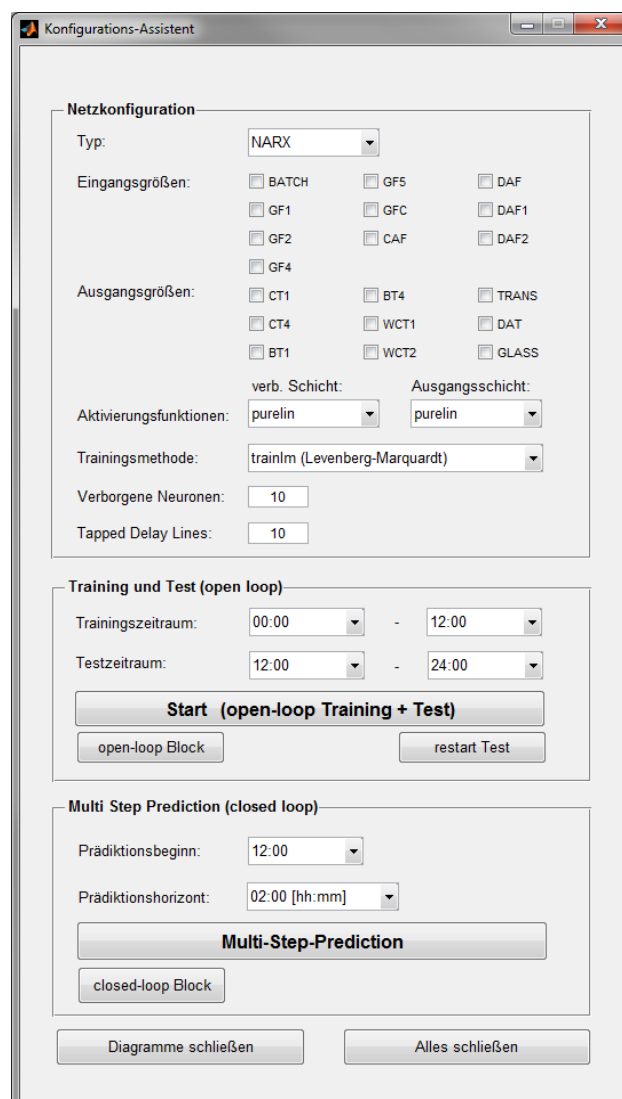


Abbildung 2.15: Grafische Benutzeroberfläche zur Konfiguration neuronaler Netze

Die Aufgabe des Konfigurationsassistenten ist die Übergabe der gewählten Netzparameter an den Quellcode zum Entwurf des neuronalen Netzes. Mit diesem Werkzeug kann der Einfluss der einzelnen Parameter schnell und vor allem strukturiert untersucht werden. Die Oberfläche ist in drei Segmente eingeteilt. Im oberen Bereich wird das neuronale Netz konfiguriert und dimensioniert. Der Netztyp (NIO, NAR, NARX) sowie die Ein- und Ausgangsgrößen werden festgelegt. Weiterhin können die Aktivierungsfunktionen in beiden Schichten, die verwendete Trainingsmethode, die Anzahl der verborgenen Neuronen und die Anzahl der Tapped Delay Lines eingegeben werden. Im zweiten Abschnitt werden die Zeiträume für das Training und den "open loop"-Test definiert (OSP). Der komplette Datensatz wird anschließend auf die ausgewählten Ein- und Ausgangsgrößen und den definierten Zeitraum reduziert. Im unteren Segment kann mit dem vorher trainierten Netz eine Multi Step Prediction durchgeführt werden, bei der Prädiktionsbeginn und -horizont frei wählbar sind. Zusätzlich ist die Möglichkeit gegeben, die trainierten KNN in Funktionsbausteine zu konvertieren, die für Simulationszwecke in *MATLAB/Simulink* genutzt werden können.

Ein weiterer Schwerpunkt bei der Programmierung des Konfigurationsassistenten war die Auswertung der Simulationsergebnisse. Das wesentliche Bewertungskriterium für das neuronale Netz ist die Genauigkeit, die anhand der berechneten Fehler für jede einzelne Ausgangsgröße bestimmt werden kann. Der Fehler wird zu jedem Zeitpunkt aus dem vom Netz berechneten Ausgangswert und dem tatsächlich gemessenen Ausgangswert berechnet. Über den gesamten Trainings-, Test- und Prädiktionszeitraum können die berechneten Fehler untersucht werden. Die Tabelle 2.2 listet die statistischen Kennzahlen auf, die zur Beurteilung der Genauigkeit in den folgenden Untersuchungen herangezogen werden.

Tabelle 2.2: Kennzahlen zur Beurteilung der Genauigkeit

Kennzahl		Bemerkung
Maximaler positiver Fehler	$e_{\max,+}$	Gibt die über den betrachteten Zeitraum größte positive Abweichung zwischen den berechneten und tatsächlichen Werten an.
Maximaler negativer Fehler	$e_{\max,-}$	Gibt die über den betrachteten Zeitraum größte negative Abweichung zwischen den berechneten und tatsächlichen Werten an.
Arithmetischer Mittelwert	$\bar{e} = \frac{1}{n} \sum_{i=1}^n e_i$	Gibt den Durchschnitt aller (n) berechneten Fehler im betrachteten Zeitraum an. Der Wert kann positiv oder negativ sein.
Standardabweichung	$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (e_i - \bar{e})^2}$	Ergibt sich aus der Wurzel der Varianz. Die Varianz ist die Summe der quadratischen Abweichungen aller (n) Fehler zum arithmetischen Mittelwert, geteilt durch die Stichprobenanzahl (n-1).

Der maximale positive und negative Fehler bilden einen Rahmen, in dem sich sämtliche Fehler im betrachteten Zeitraum befinden. Die Genauigkeit kann anhand dieser beiden Kennzahlen nicht vollständig beurteilt werden. Der arithmetische Mittelwert liefert eine Aussage, ob der berechnete Verlauf unter oder über den tatsächlichen Werten liegt. Für eine aussagekräftige Beurteilung der Genauigkeit sind zusätzlich Kennzahlen zur Streuung notwendig, z. B. die Varianz oder die Standardabweichung. Die Varianz gibt Auskunft darüber, wie verstreut die Fehler vom Mittelwert aller Fehler liegen. Die Varianz liefert quadratische Einheiten, die eine Interpretation der Ergebnisse erschweren. Die Standardabweichung entspricht der Wurzel der Varianz und liefert somit physikalisch verständlichere Einheiten. Die Kombination aller vier Kennzahlen erlaubt eine Aussage über die Genauigkeit des trainierten Netzes. Um eine Aussage zum Rechenaufwand treffen zu können werden zusätzlich die benötigten Rechenzeiten und die Anzahl der Iterationen während des Trainings ausgewertet. Als weiteres Hilfsmittel zur Bewertung der Simulationsergebnisse erstellt der Konfigurationsassistent automatisch generierte Diagramme, in denen die Verläufe der tatsächlichen (Targets) und berechneten Ausgangswerte (Outputs) im Trainings- und Testzeitraum sowie deren berechnete Fehler gegenübergestellt werden. Abbildung 2.16 zeigt ein exemplarisches Diagramm für die Ausgangsgröße WCT1.

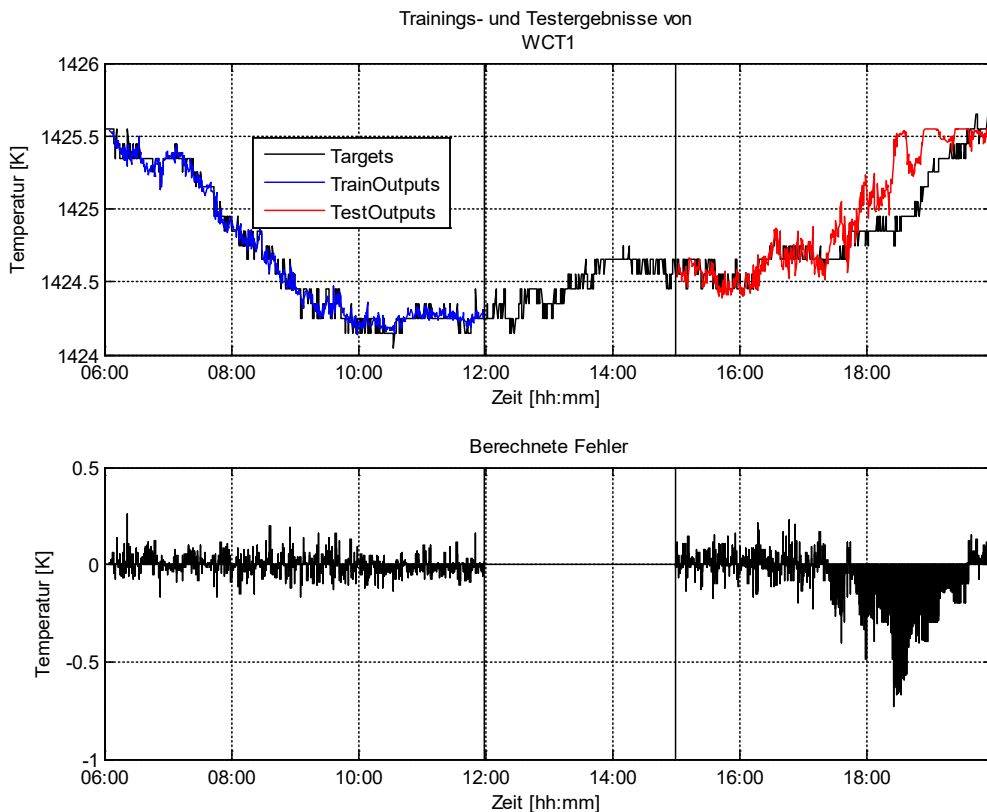


Abbildung 2.16: Beispielhaftes Ausgabediagramm des Konfigurationsassistenten

---

## 3 Identifikation geeigneter neuronaler Prozessmodelle

Unter einem geeigneten neuronalen Prozessmodell wird ein Künstliches Neuronales Netz verstanden, das mit Hilfe gegebener Messdaten der Eingangsgrößen mindestens eine der Ausgangsgrößen hinreichend genau berechnen kann. Der Fokus liegt dabei auf den qualitätsbestimmenden Temperaturen in der Abstehtwanne (WCT1, WCT2, TRANS) und dem Glasstand GLASS. Die Schwierigkeit besteht in der Komplexität der Konfigurationsmöglichkeiten. Der erste Abschnitt erläutert die Herangehensweise zur Identifikation. Die Simulationen haben unterschiedliche Einflüsse der einzelnen Netzparameter auf die Genauigkeit der berechneten Ausgangsgrößen aufgezeigt. Eine ausführliche Auswertung der Simulationsergebnisse wird im zweiten Abschnitt gegeben, bevor im Anschluss die Erkenntnisse der Prozessidentifikation zusammengefasst werden.

### 3.1 Methodik

Die einstellbaren Parameter wurden nach ihrer Relevanz geordnet und schrittweise geändert, um eine nachvollziehbare Herangehensweise zu gewährleisten. Die vollständige Simulationsumgebung ist in Tabelle 3.1 aufgelistet. Der Einfluss der einzelnen Netzparameter ist für alle drei Netztypen (NIO, NAR, NARX) untersucht worden. Die Netze wurden open loop trainiert und getestet.

Die variable Simulationsumgebung gliedert sich in sechs Etappen. Im ersten Test wurden 20 verschiedene Kombinationen von Eingangsgrößen untersucht. Bei der Auswahl sind prozessnahe Aspekte, vor allem die lokale Anordnung der Messstellen, berücksichtigt worden. Es wurde zusätzlich untersucht, inwieweit sich die Ergebnisse unterscheiden, wenn das Netz nur eine Ausgangsgröße (Multiple Input Single Output, MISO) oder alle vier Ausgangsgrößen (Multiple Input Multiple Output, MIMO) besitzt.

Im zweiten Test sind die verfügbaren Trainingsmethoden Untersuchungsgegenstand. Dabei wurden die Netze verwendet, die sich im ersten Test als geeignet herausstellten. Diese Herangehensweise wird für die folgenden Schritte fortgesetzt. Die Ergebnisse der vorangegangenen Untersuchung sind in der darauffolgenden Simulation berücksichtigt.

Tabelle 3.1: Simulationsumgebung und Testreihenfolge

Konstante Simulationsumgebung:			
Benutzeroberfläche	Konfigurationsassistent (siehe Abbildung 2.15)		
Netztyp	1) NARX	2) NIO	3) NAR
Architektur	Training: open loop		Test: open loop
Datensatz	linear interpoliert, Schrittweite 30 Sekunden		
Arbeitsspeicher	16 GB		
Prozessor	Intel Core i5-4690, 3.5 GHz, 4 Kerne		
Betriebssystem	Windows 7		
Software	MATLAB R2014a		
Variable Simulationsumgebung:			
Test 1:	<b>Ein- und Ausgangsgrößen</b> Eingangsgrößen: 20 verschiedene Kombinationen Ausgangsgrößen: WCT1, WCT2, TRANS, GLASS Strukturen: SISO, MISO, MIMO		
Test 2:	<b>Trainingsmethode</b> 12 verfügbare Methoden (Default: Levenberg-Marquardt-Algorithmus)		
Test 3:	<b>Aktivierungsfunktionen in beiden Schichten</b> 15 verfügbare Funktionen (Default: <i>tansig</i> / <i>purelin</i> )		
Test 4:	<b>Anzahl verborgener Neuronen</b> Änderung von 1 bis 100 Neuronen, Schrittweite: +1 (Default: 10)		
Test 5:	<b>Anzahl Tapped Delay Lines</b> Änderung von 1 bis 120 TDL, Schrittweite: +1 (Default: 10)		
Test 6:	<b>Trainings- und Testzeiträume</b> Länge vom Trainingszeitraum: 1 - 12 Stunden Länge vom Testzeitraum: 1 - 12 Stunden (Default: 00:00 - 12:00 Uhr Training, 12:00 - 24:00 Test)		

## 3.2 Simulationsergebnisse

### 3.2.1 Auswahl der Ein- und Ausgangsgrößen

Die Auswertung gliedert sich in

- (a) NARX-Netze, die alle vier Ausgangsgrößen gleichzeitig berechnen,
- (b) NARX-Netze, die nur eine der vier Ausgangsgrößen berechnen,
- (c) NIO-Netze und
- (d) NAR-Netze.

Für (a) sind nachfolgend die drei besten Eingangskombinationen aufgelistet.

- Netz A, NARX: DAF
- Netz B, NARX: BATCH, GFC, DAF
- Netz C, NARX: BATCH, GFC, CAF, DAF

Mit der Gemeindegabe BATCH, der Gesamtgasmenge GFC, der Verbrennungsluftmenge CAF und der Kühlluftmenge DAF werden die wichtigsten Stellgrößen einbezogen. Die Erwartung, dass mehr Eingangsgrößen aufgrund des größeren Informationsgehaltes zu einer erhöhten Genauigkeit führen, kann nicht bestätigt werden. Es zeigte sich, dass mehrere Eingangsgrößen einen erhöhten Rechenaufwand fordern, was in längeren Trainingszeiten und numerischen Ungenauigkeiten resultierte. Je mehr Eingangsgrößen verwendet werden, desto ungenauer die Ergebnisse.

Die Netze A, B und C berechnen die vier Ausgangsgrößen WCT1, WCT2, TRANS und GLASS gleichzeitig. In der NARX-Struktur bedeutet dies, dass z. B. zur Berechnung von WCT2 auch die Messwerte der anderen Ausgangsgrößen herangezogen werden. Es kann somit keine eindeutige Aussage getroffen werden, inwieweit sich das Ergebnis aus den Ein- oder Ausgangsgrößen zusammensetzt. Für die Ausgangsgröße WCT2 (Netz C) beträgt die Standardabweichung im Testzeitraum 0,3872 K. Wird Netz C so konfiguriert, dass es die Temperatur WCT2 als einzige Ausgangsgröße (MISO) besitzt, verringert sich die Standardabweichung um 64 % auf 0,1821 K. Abbildung 3.1 verdeutlicht diesen Sachverhalt.

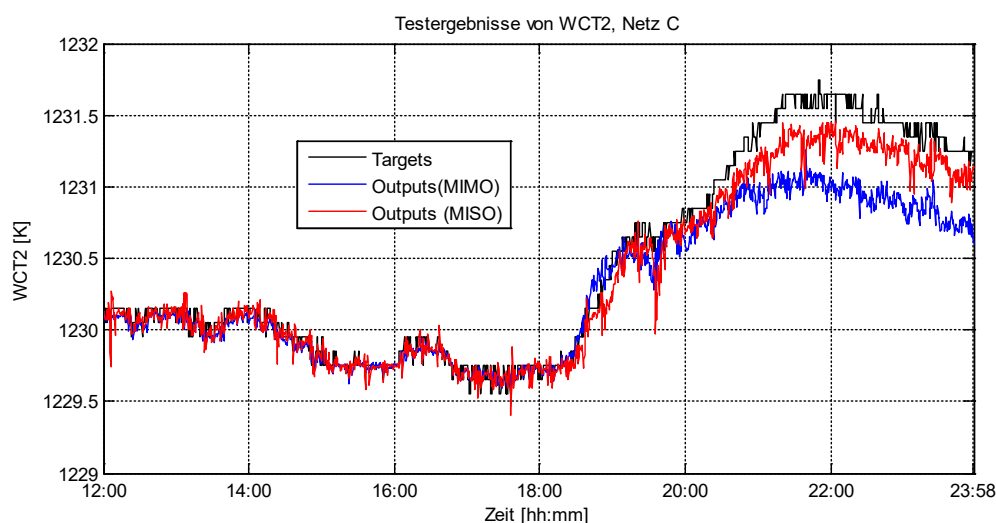


Abbildung 3.1: Vergleich zwischen der einzelnen (MISO) und gleichzeitig (MIMO) berechneten Ausgangsgröße WCT2



Nachfolgend sind für (b) drei NARX-Netze mit ihren Standardabweichungen  $S$  im Testzeitraum aufgelistet, mit denen sich annehmbare Ergebnisse erzielen lassen.

- Netz D, NARX: BATCH / GLASS ( $S = 0,0309$  mm)
- Netz E, NARX: GFC, DAF / WCT1 ( $S = 0,0551$  K)
- Netz F, NARX: BATCH, GFC, CAF, DAF / WCT2 ( $S = 0,1052$  K)

Im Vergleich zum Testzeitraum sind die Simulationsergebnisse im Trainingszeitraum erwartungsgemäß besser. Bei der Ausgangsgröße GLASS steigt die Standardabweichung im Testzeitraum um 4 %, bei Netz F sogar um 116 %.

KNN vom Typ NIO (c) sind für die Berechnung von Prozessgrößen in einer Flachglaswanne nicht geeignet. In den meisten Fällen erzeugte das trainierte Netz stark verrauschte Signale oder konstante Verläufe. Lediglich bei einem Netz konnte der Verlauf der Ausgangsgröße tendenziell abgebildet werden.

- Netz G, NIO: BATCH / GLASS ( $S = 0,0745$  mm)

Im Vergleich zum NARX-Netz D ist die Standardabweichung deutlich höher, dennoch zeigt Abbildung 3.2 ein vielversprechendes Testergebnis. Es muss berücksichtigt werden, dass Netz G allein anhand der vergangenen zehn Messwerte (TDL = 10) der Gemengezufuhr BATCH den Glasstand GLASS berechnet. Hier ist eine Erhöhung der Genauigkeit durch eine Erhöhung des TDL-Wertes zu erwarten (Abschnitt 3.2.5).

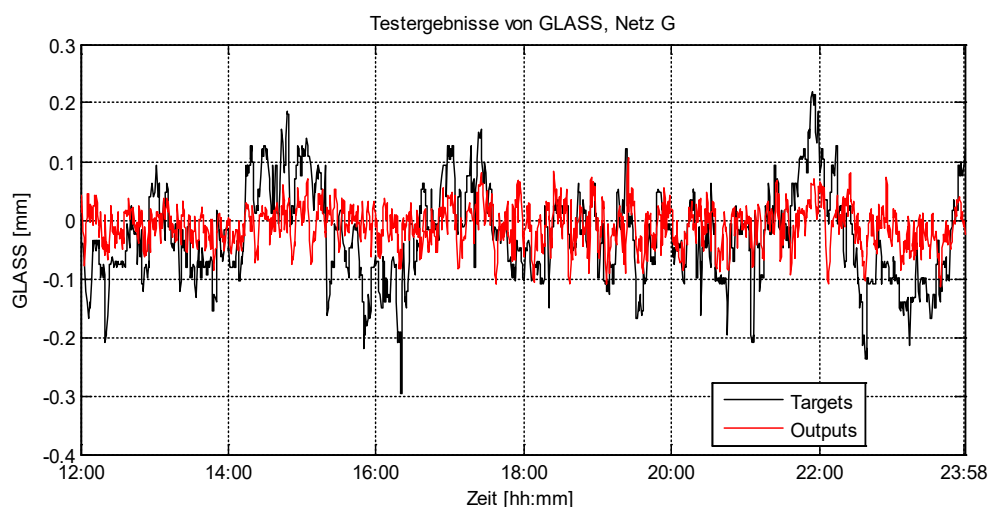


Abbildung 3.2: Testergebnis von Netz G zur Berechnung des Glasstandes

Die Simulationsergebnisse zu (d) sind sehr gut. Einzelne Ausgangsgrößen können in der NAR-Struktur noch genauer berechnet werden als in der NARX-Struktur. Nachfolgend sind die drei besten NAR-Netze und deren Standardabweichungen im Testzeitraum aufgelistet.

- Netz H, NAR:           WCT1                   (S = 0,0499 K)
- Netz I, NAR:           WCT2                   (S = 0,0653 K)
- Netz J NAR:           GLASS                   (S = 0,0305 K)

Abbildung 3.3 zeigt die Trainings- und Testergebnisse für Netz I. Im Vergleich zu Abbildung 3.1 kann besonders im Testzeitraum der tatsächliche Verlauf deutlich besser angenähert werden. Die logische Schlussfolgerung ist, dass die berechnete Ausgangsgröße, insbesondere auch bei NARX-Netzen, größtenteils von vergangenen Werten dieser Ausgangsgröße abhängig ist. Der Einfluss der angelegten Eingangsgrößen kann bei NARX-Netzen somit nicht genau bestimmt werden.

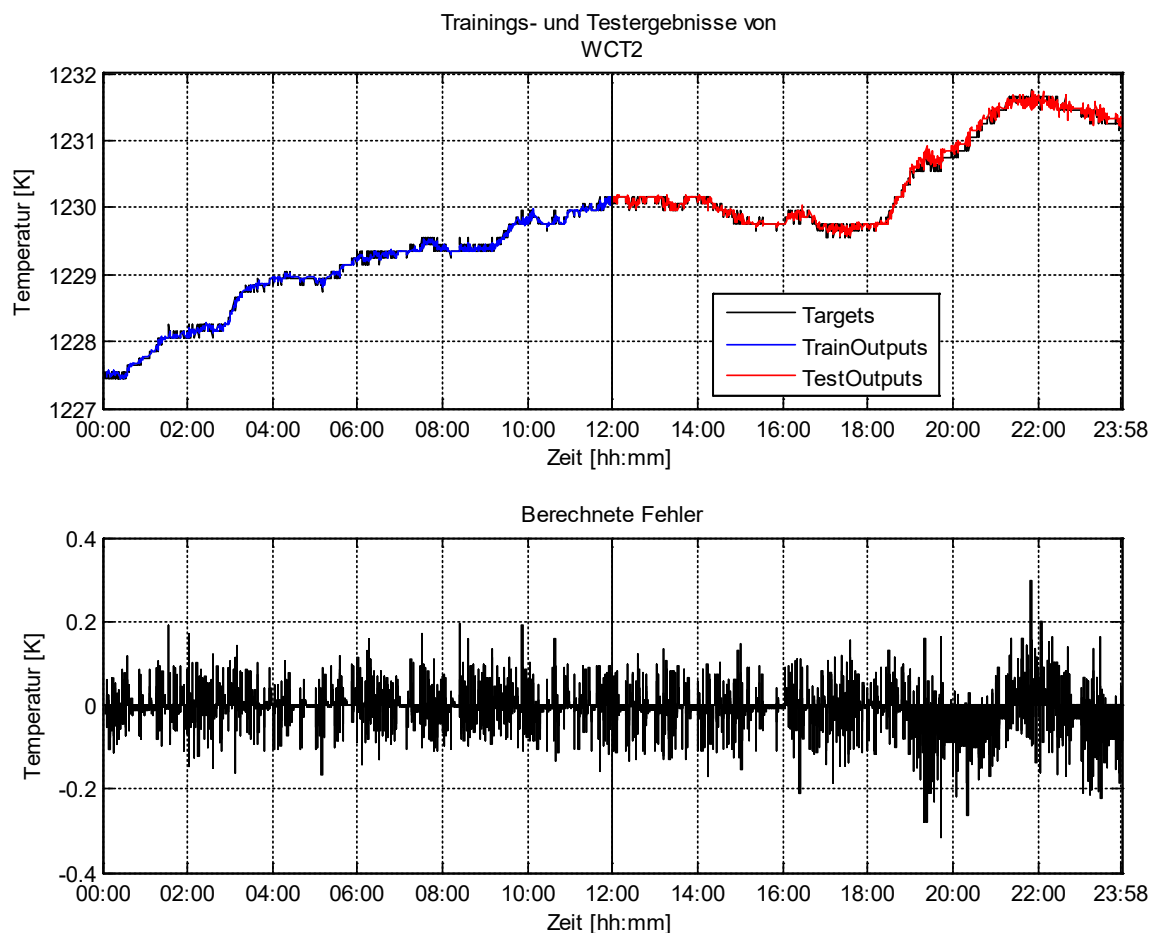


Abbildung 3.3: Trainings- und Testergebnisse vom NAR-Netz I

Für die weiterführenden Tests werden folgende fünf Netze berücksichtigt (Abbildung 3.4).

	<u>Eingangsgröße:</u>	<u>Ausgangsgröße:</u>	<u>Typ:</u>
▪ <b>Netz 1:</b>	BATCH, GFC, CAF, DAF	WCT2	NARX
▪ <b>Netz 2:</b>	GFC, DAF	WCT1	NARX
▪ <b>Netz 3:</b>	BATCH	GLASS	NARX
▪ <b>Netz 4:</b>	BATCH	GLASS	NIO
▪ <b>Netz 5:</b>		WCT2	NAR

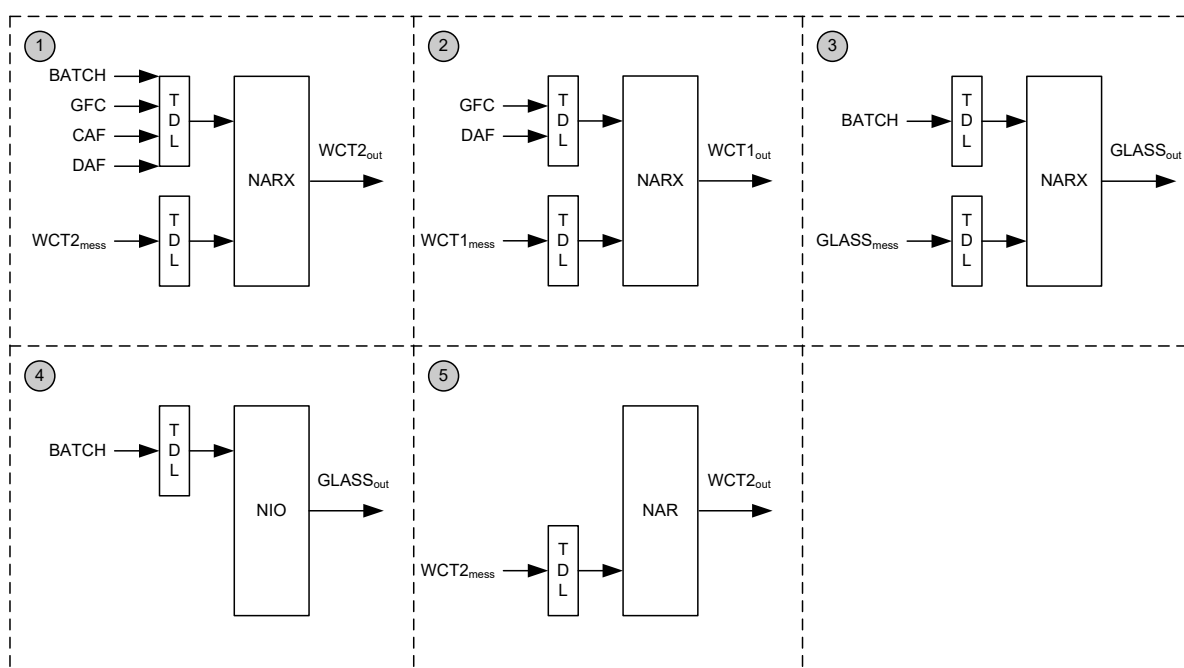


Abbildung 3.4: Übersicht der fünf evaluierten Netze

### 3.2.2 Einfluss der Trainingsmethoden

Die Wahl der Trainingsmethode hat einen maßgeblichen Einfluss auf die Genauigkeit der berechneten Ausgangsgröße. In Abbildung 3.5 sind für alle zwölf verfügbaren Trainingsmethoden die zeitlichen Verläufe der Ausgangsgröße WCT2 (Netz 1) im Testzeitraum dargestellt. Aus Gründen der Übersichtlichkeit wurde auf die Achsenbeschriftung verzichtet. Auf der Ordinate ist der Temperaturwert in Kelvin und auf der Abszisse die Zeit angegeben. Der Titel des jeweiligen Diagramms kennzeichnet die verwendete Trainingsmethode. Für Netz 1 kommen zwei Trainingsmethoden (rot hervorgehoben) infrage.

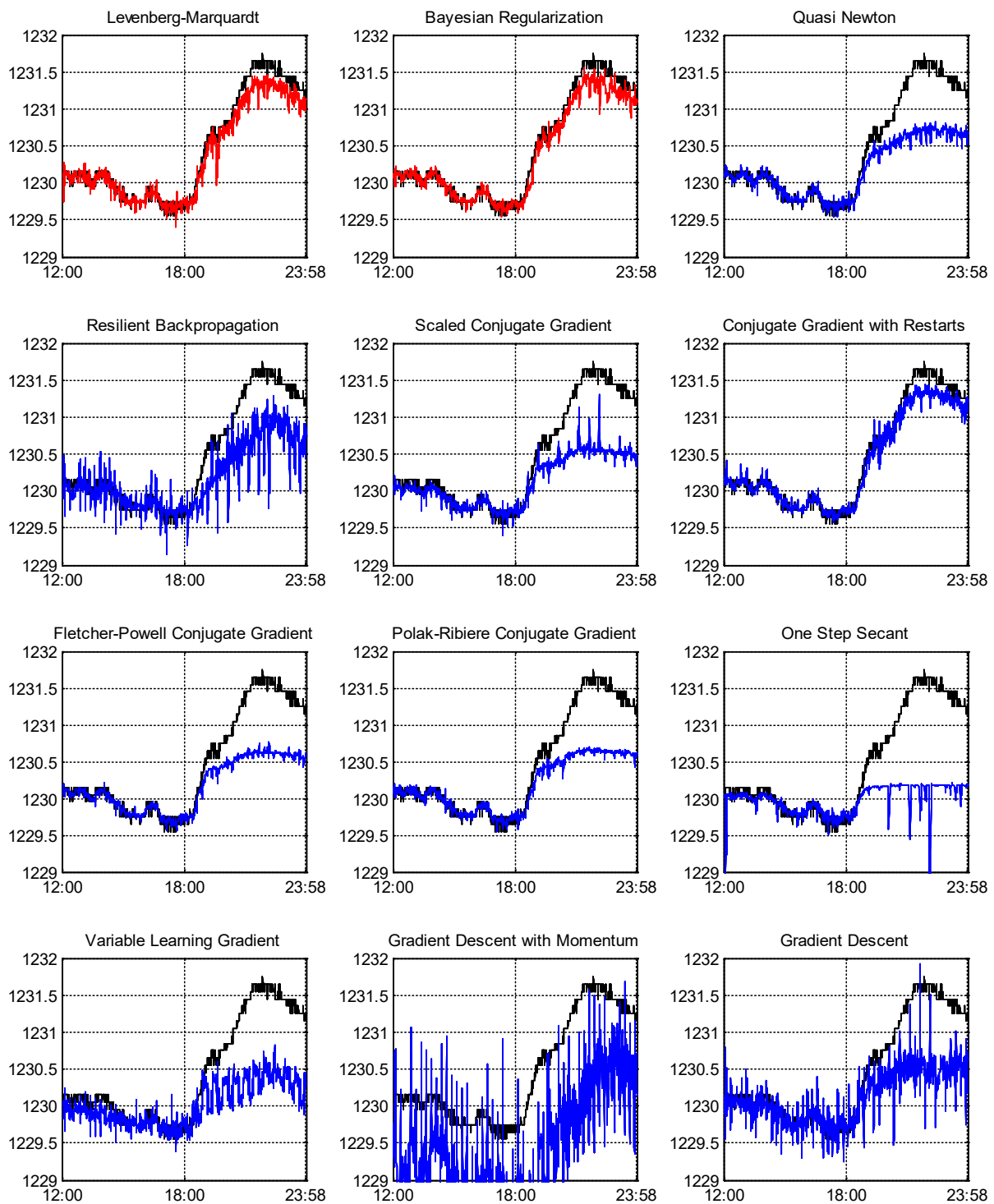


Abbildung 3.5: Zeitliche Verläufe der tatsächlichen (schwarz dargestellt) und berechneten Ausgangsgröße WCT2 (Netz1) im Testzeitraum für verschiedene Trainingsmethoden (Bem.: Auf der Abszisse ist die Zeit [hh:mm], auf der Ordinate die Temperatur [K] dargestellt. Die verwendete Trainingsmethode ist Titel des Diagramms.)

Nachfolgend sind für die fünf Netze die zwei Trainingsmethoden aufgelistet, mit denen die geringsten Standardabweichungen erreichbar sind.

- Netz 1:        Levenberg-Marquardt        Bayesian Regularization
- Netz 2:        Levenberg-Marquardt        Bayesian Regularization
- Netz 3:        Levenberg-Marquardt        Fletcher-Powell Conjugate Gradient
- Netz 4:        Quasi Newton                      Resilient Backpropagation
- Netz 5:        Levenberg-Marquardt        Quasi Newton

Neben den zeitlichen Verläufen sind die Anzahl der Iterationen und die benötigte Trainingszeit zur Auswertung herangezogen worden. In Tabelle 3.2 sind alle Trainingsmethoden nach diesen Kriterien aufsteigend geordnet. Zusätzlich sind die Standardabweichungen gegeben. Bei allen Werten handelt es sich um den Mittelwert aller fünf Netze, weshalb bei der Standardabweichung keine Einheit angegeben ist. Die ausführlichen Bezeichnungen der Funktionsnamen sind in Anhang A und weiterführende Informationen in [67] gegeben.

Tabelle 3.2: Ranking der verschiedenen Trainingsmethoden

Rank	Anzahl Iterationen		Trainingszeit [sec]		Standardabweichung	
	Funktion	Durchschnitt	Funktion	Durchschnitt	Funktion	Durchschnitt
1	trainlm	9	trainlm	0,23	trainbr	0,0660
2	traincgp	48	traincgp	4,68	trainlm	0,0810
3	traincgb	49	trainoss	4,87	traincgb	0,1078
4	trainbfg	54	traincgb	5,04	trainscg	0,1144
5	traincgf	57	traingdx	5,24	traincgf	0,1186
6	trainoss	58	trainscg	5,41	trainbfg	0,1268
7	trainscg	75	trainrp	5,52	trainoss	0,1373
8	traingdx	138	traincgf	5,56	traincgp	0,1464
9	trainrp	143	trainbfg	5,64	traingdx	0,1607
10	traingdm	405	traingdm	14,46	trainrp	0,1646
11	trainbr	672	trainbr	17,51	traingd	0,1751
12	traingd	1.000	traingd	36,70	traingdm	0,3559

Die Levenberg-Marquardt-Methode (*trainlm*) ist deutlich schneller als die anderen verfügbaren Methoden. Neun Iterationen im Training stellen mit Abstand den Bestwert dar. Eine höhere Genauigkeit, gemessen anhand der Standardabweichung, wird nur mit dem Bayesian Regularization Algorithmus (*trainbr*) erreicht. Wie in Abbildung 3.5 zu sehen, ist der Unterschied der beiden Methoden nur gering. Die deutlich längeren Trainingszeiten der

Bayesian Regularization Methode sind ein nicht zu unterschätzender Nachteil. Die Schrittweite des Datensatzes beträgt 30 Sekunden, die Trainingszeiten im Durchschnitt 17,51 Sekunden, für Netz 1 sogar über 36 Sekunden. In einer prädiktiven Regelstrategie, in der eine numerische Optimierung, ähnlich wie beim Training der KNN, durchgeführt werden muss, können längere Rechenzeiten zu Problemen bei der Echtzeitausführung führen. Aus diesem Grund wird der Algorithmus von Levenberg-Marquardt empfohlen.

#### 3.2.3 Einfluss der Aktivierungsfunktionen

Die Aktivierungsfunktionen haben entscheidenden Einfluss auf die Netzcharakteristik. Die verfügbaren Funktionen sind im Anhang B beschrieben. Die Aktivierungsfunktion kann für die verborgene Schicht und die Ausgangsschicht festgelegt werden. Es wurden alle 225 Kombinationen getestet. Neben den berechneten statistischen Kennzahlen dienten vor allem die zeitlichen Verläufe der berechneten Ausgangsgrößen zur Auswertung. Nachfolgend sind in Abbildung 3.6 die Trainings- und Testergebnisse von Netz 1 (WCT2) für alle verfügbaren Aktivierungsfunktionen dargestellt. Die ausgewählte Funktion wurde in beiden Schichten verwendet. Aus Gründen der Übersichtlichkeit ist erneut keine Achsenbeschriftung gegeben. Netz 1 ist für die Darstellung der Ergebnisse geeignet, da der Wertebereich im Testzeitraum größer ist als im Trainingszeitraum.

Der Abbildung 3.6 ist zu entnehmen, dass lediglich die Aktivierungsfunktion *purelin* für Netz 1 geeignet ist. Die Funktionen mit Zweipunktverhalten (*hardlim*, *hardlims*) sowie die Radial-Basisfunktionen (*tribas*, *radbas*, *radbasn*) liefern durchweg inakzeptable Ergebnisse. Gleiches gilt für die Aktivierungsfunktionen *compet*, *softmax* und *netinv*.

Die Tauglichkeit der begrenzenden Aktivierungsfunktionen hängt maßgeblich vom Wertebereich der Ausgangsgröße ab. Zu den begrenzenden Aktivierungsfunktionen gehören die linearen (*poslin*, *satlin*, *satlins*) und die sigmoiden (*tansig*, *logsig*, *elliotsig*) Funktionen. Bei *poslin*, *satlin* und *logsig* werden keine negativen Werte ausgegeben. In Abbildung 3.6 ist das dadurch erkennbar, dass diese Netze keine Ausgangsgrößen erzeugen, die kleiner der Hälfte des Trainingswertebereichs sind.

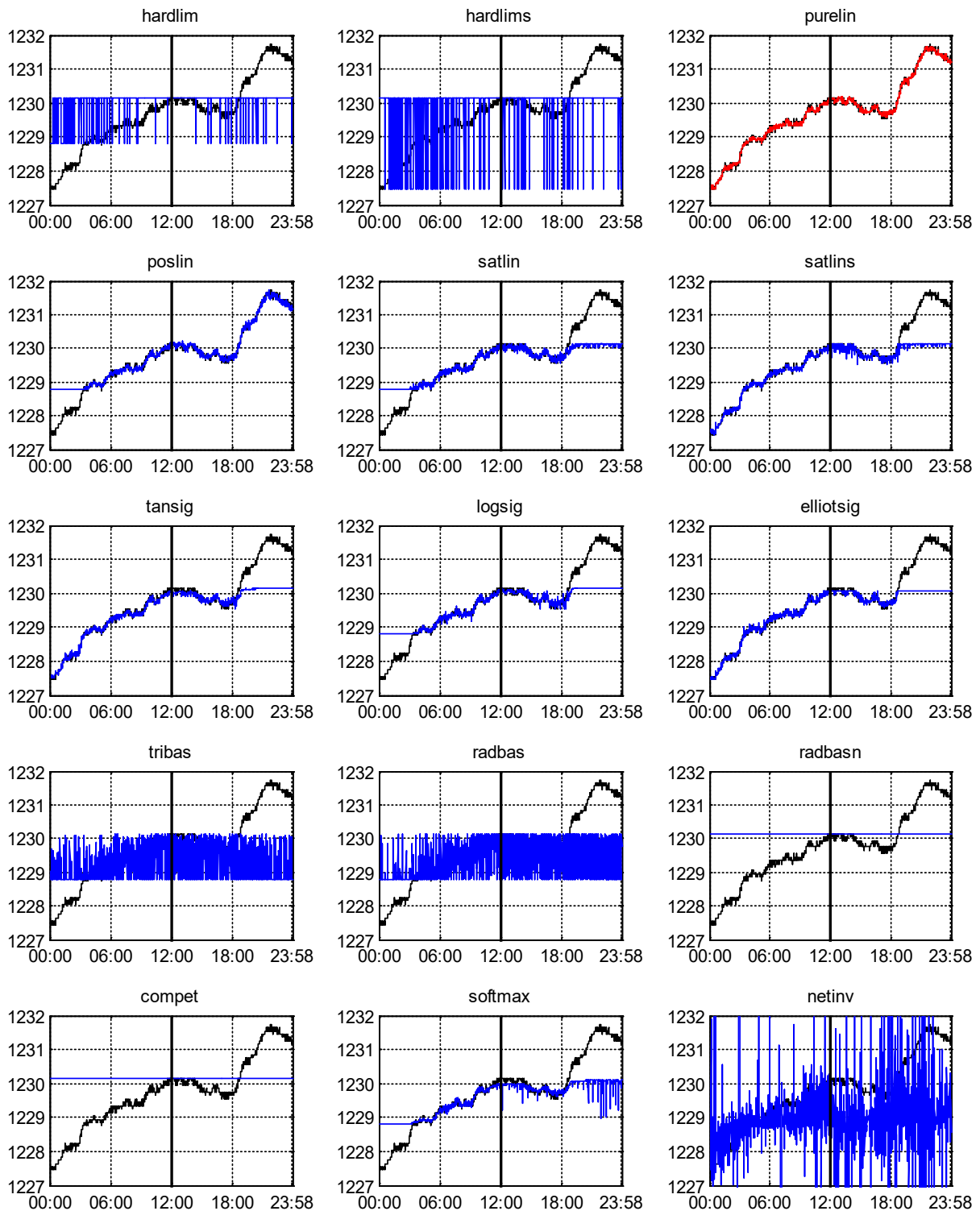


Abbildung 3.6: Simulationsergebnisse von Netz 1 (WCT2) im Trainings- und Testzeitraum für verschiedene Aktivierungsfunktionen (Bem.: Auf der Abszisse ist die Zeit [hh:mm], auf der Ordinate die Temperatur [K] dargestellt. Die in beiden Schichten verwendete Aktivierungsfunktion ist Titel des Diagramms.)

Bei den Funktionen *satlins*, *tansig* und *elliotsig* kann das Ausgangssignal negative Werte annehmen, das heißt es wird der gesamte Trainingswertebereich abgedeckt. Liegt die zu berechnende Ausgangsgröße außerhalb dieses Bereiches, sind auch diese Funktionen nicht mehr geeignet. Bei den Aktivierungsfunktionen mit Sättigung ist demzufolge darauf zu achten, dass das Netz in einem sinnvollen und vollständigen Wertebereich trainiert wird. Abbildung 3.7 verdeutlicht diesen Sachverhalt anhand der Aktivierungsfunktion *satlins* am Beispiel der NARX-Netze 1 und 3. Bei Netz 3 wird der gesamte Wertebereich im Training abgedeckt, weshalb im Gegensatz zu Netz 1 ein annehmbares Testergebnis erzielt wird.

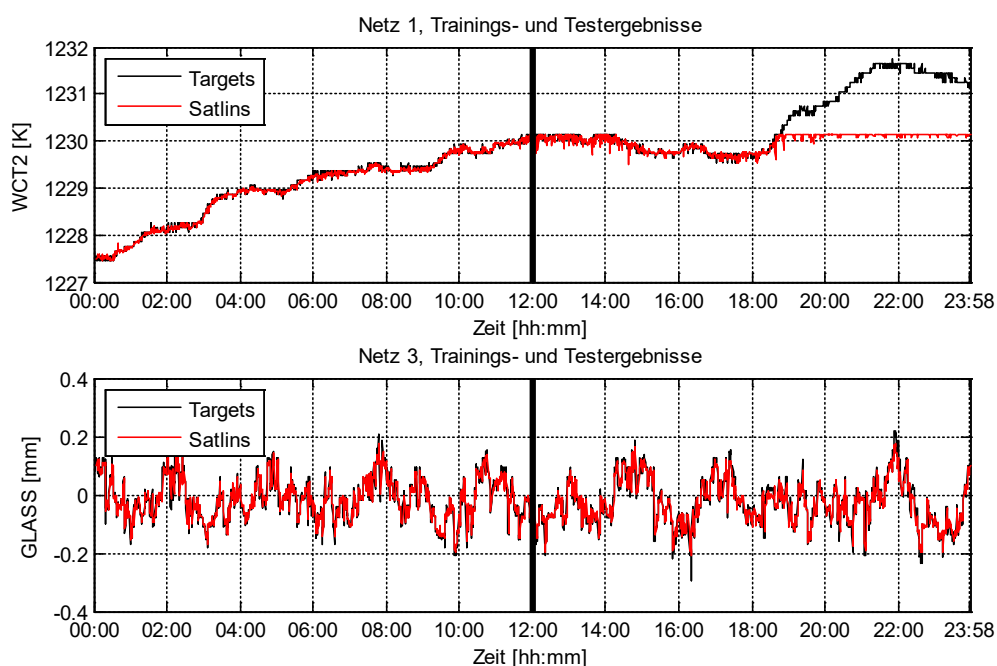


Abbildung 3.7: Vergleich der Trainings- und Testergebnisse der Netze 1 und 3 bei Verwendung der Aktivierungsfunktion *satlins*

Es lässt sich schlussfolgern, dass begrenzende Aktivierungsfunktionen für diesen Anwendungsfall ungeeignet sind. Verlässt die Prozessgröße den trainierten Wertebereich, liefert das neuronale Prozessmodell zwangsläufig fehlerhafte Ausgangswerte, die in einer prädiktiven Regelstrategie zu kontraproduktiven Stelleingriffen führen können.

Die Begrenzungseigenschaft hat bei sensiblen und turbulenten Prozessen mit minimalen Zeitverzögerungen den Vorteil, dass eventuelle Ausreißer oder Messrauschen unterdrückt werden. Die für die vorliegende Aufgabenstellung relevanten Prozessgrößen unterliegen nur geringfügigen Änderungen. Ausreißer oder übermäßige Schwankungen sind für diese Aus-



gangsgrößen aus physikalischer Sicht nahezu ausgeschlossen. Aus diesem Grund kann auf die Begrenzungsseigenschaft verzichtet werden.

Die bisherigen Erkenntnisse beziehen sich auf Netze, die in beiden Schichten die gleiche Aktivierungsfunktion nutzen. Zur Auswertung aller weiteren Kombinationen wurde für jedes Netz eine Kombinationsmatrix aufgestellt (Beispiel Tabelle 3.3 für Netz 3). Die Matrix enthält die zu jeder Kombination berechnete Standardabweichung im Testzeitraum. Der Bestwert ist grün, alle nicht geeigneten Kombinationen sind dunkelgrau schattiert.

Tabelle 3.3: Kombinationsmatrix der Aktivierungsfunktionen, Testergebnisse für Netz 3 (Bem.: die Standardabweichungen sind in [mm] angegeben)

Netz 3		Aktivierungsfunktion in der verborgenen Schicht														
		hardlim	hardlims	purelin	poslin	satlin	satlins	tansig	logsig	elliotsig	tribas	radbas	radbasn	compet	softmax	netinv
Aktivierungsfunktion in der Ausgangsschicht	hardlim	0,084	0,103	0,136	0,118	0,084	0,136	0,096	0,078	0,133	0,094	0,150	0,107	0,095	0,087	0,121
	hardlims	0,171	0,202	0,255	0,182	0,182	0,187	0,159	0,148	0,169	0,099	0,195	0,082	0,125	0,082	0,227
	purelin	0,060	0,074	0,029	0,032	0,033	0,031	0,031	0,031	0,034	0,037	0,036	0,033	0,067	0,033	0,168
	poslin	0,082	0,081	0,059	0,060	0,061	0,060	0,061	0,059	0,061	0,061	0,082	0,061	0,079	0,082	0,143
	satlin	0,069	0,076	0,059	0,060	0,060	0,059	0,060	0,060	0,058	0,059	0,061	0,060	0,074	0,060	0,080
	satlins	0,053	0,056	0,029	0,032	0,030	0,030	0,031	0,030	0,032	0,033	0,033	0,036	0,062	0,032	0,083
	tansig	0,065	0,068	0,030	0,031	0,032	0,032	0,032	0,032	0,032	0,035	0,035	0,033	0,061	0,032	0,082
	logsig	0,074	0,068	0,060	0,059	0,059	0,060	0,059	0,060	0,058	0,060	0,061	0,064	0,075	0,060	0,101
	elliotsig	0,055	0,066	0,033	0,035	0,035	0,035	0,034	0,034	0,031	0,048	0,039	0,035	0,066	0,034	0,084
	tribas	0,078	0,080	0,084	0,072	0,060	0,081	0,065	0,059	0,082	0,082	0,075	0,060	0,082	0,062	0,088
	radbas	0,076	0,071	0,082	0,086	0,060	0,059	0,066	0,073	0,063	0,063	0,062	0,061	0,071	0,061	0,085
	radbasn	0,082	0,082	0,082	0,082	0,082	0,082	0,082	0,082	0,082	0,082	0,082	0,082	0,082	0,082	0,082
	compet	0,082	0,082	0,082	0,082	0,082	0,082	0,082	0,082	0,082	0,082	0,082	0,082	0,082	0,082	0,082
	softmax	0,075	0,074	0,060	0,060	0,059	0,060	0,060	0,061	0,060	0,060	0,061	0,061	0,076	0,059	0,074
	netinv	0,101	0,074	0,089	0,519	19,29	0,197	0,136	0,230	0,376	0,069	0,124	0,064	0,066	0,053	0,454

Nachfolgend sind für alle fünf Netze die besten Kombinationen aufgelistet.

	<u>Schicht 1</u>	<u>Schicht 2</u>	<u>Standardabweichung</u>
▪ Netz 1:	<i>purelin</i>	<i>purelin</i>	0,0533 K
▪ Netz 2:	<i>purelin</i>	<i>purelin</i>	0,0506 K
▪ Netz 3:	<i>purelin</i>	<i>satlins</i>	0,0291 mm
▪ Netz 4:	<i>tansig</i>	<i>elliotsig</i>	0,0740 mm
▪ Netz 5:	<i>purelin</i>	<i>purelin</i>	0,0505 K

Die Kombination *purelin* / *purelin* führt bei drei von fünf Netzen zur höchsten Genauigkeit. Auch bei Netz 3 ( $S = 0,0292$  mm) und Netz 4 ( $S = 0,0773$  mm) sind die Ergebnisse mit dieser Kombination nur marginal schlechter als die jeweils beste Kombination. Aus diesem Grund wird in beiden Schichten die lineare Aktivierungsfunktion *purelin* empfohlen.

### 3.2.4 Einfluss der verborgenen Neuronen

Eine Erhöhung der Anzahl verborgener Neuronen in Schicht 1 hat bei Verwendung der linearen Aktivierungsfunktion keinen nennenswerten Einfluss auf die Berechnung der Ausgangsgröße. In Abbildung 3.8 sind für den Testzeitraum die berechneten Standardabweichungen gegen die Neuronenanzahl aufgetragen.

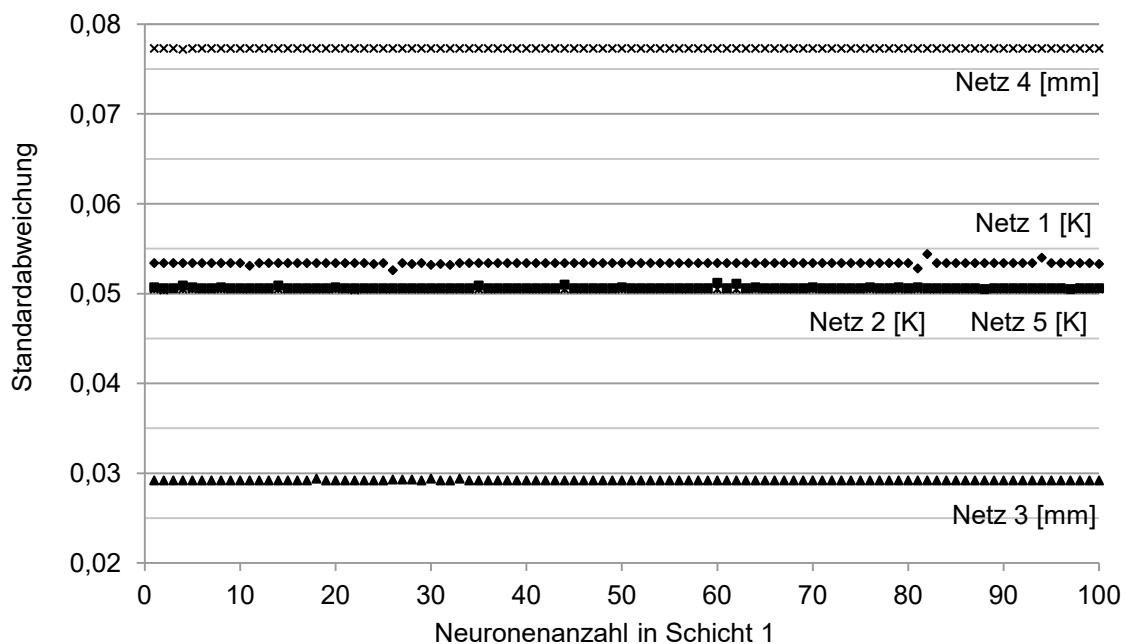


Abbildung 3.8: Standardabweichungen in Abhängigkeit der Neuronenanzahl bei linearer Aktivierungsfunktion

Eine zusätzliche Durchführung der Simulation mit der standardmäßig eingestellten Kombination *tansig* / *purelin* führt zu einem anderen Ergebnis (Abbildung 3.9). Aufgrund der unterschiedlichen Wertebereiche aller fünf Netze wurden die Standardabweichungen normiert. Unabhängig vom verwendeten Netz ist erkennbar, dass die Genauigkeit des Netzes mit steigender Neuronenanzahl abnimmt. Bei höheren Neuronenzahlen nimmt die Schwankungsbreite der zeitlichen Verläufe signifikant zu.

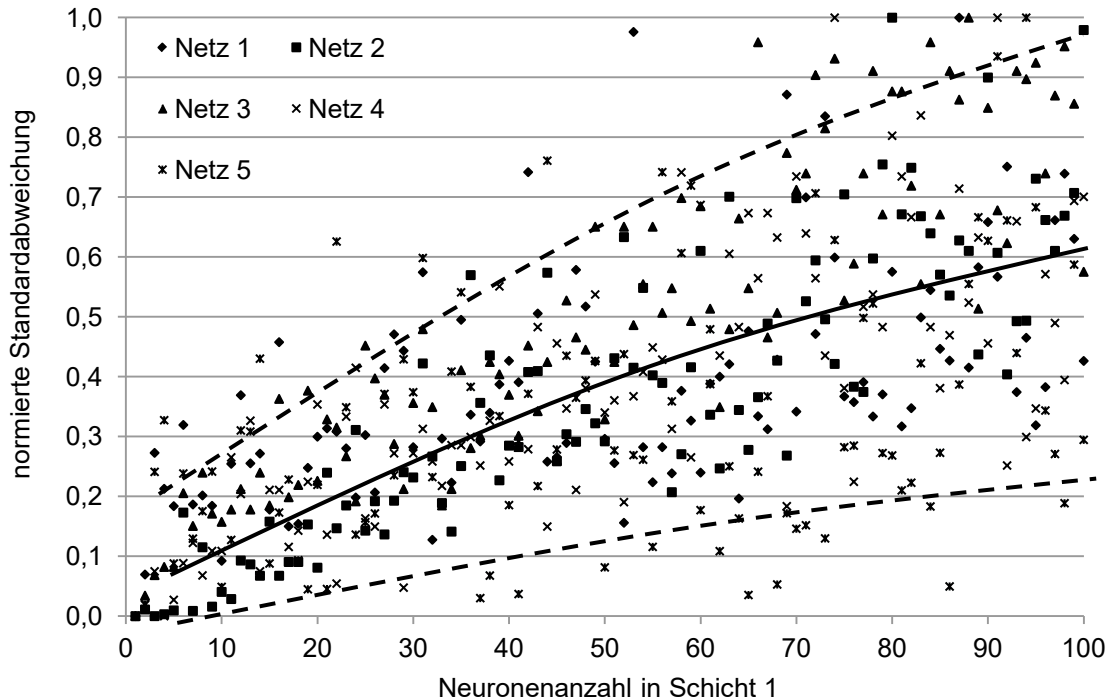


Abbildung 3.9: Normierte Standardabweichungen in Abhängigkeit der Neuronenanzahl bei sigmoider Aktivierungsfunktion

Der Einfluss der verborgenen Neuronen richtet sich demzufolge nach der verwendeten Aktivierungsfunktion. Zur weiteren Veranschaulichung gibt Abbildung 3.10 eine direkte Gegenüberstellung der berechneten zeitlichen Verläufe am Beispiel der Ausgangsgröße WCT2 (Netz 1). Die Anzahl der verwendeten Neuronen und die Aktivierungsfunktionen sind Titel der jeweiligen Diagramme. Die Verwendung der Aktivierungsfunktion *tansig* führt zu einer sichtbar schlechteren Qualität des Ausgangssignals.

Die Ursache kann allein auf die begrenzende Eigenschaft der sigmoiden Tangensfunktion und den daraus resultierenden Rechenaufwand zurückgeführt werden. Im Anhang E ist eine beispielhafte Berechnung eines konkreten Temperaturwertes für WCT1 mit verschiedenen Neuronenzahlen und Aktivierungsfunktionen gegeben. Es zeigen sich erhebliche Unterschiede in den Werten einzelner Komponenten, besonders im Schwellwertvektor und in der Wichtungsmatrix  $LW^{2,1}$ .

Die Verwendung der *tansig*-Funktion führt zu einem erhöhten Rechenaufwand. In Abbildung 3.11 sind für die Netze 1 und 3 die Trainingszeiten gegen die Neuronenanzahl aufgetragen. Die gemessenen Trainingszeiten lassen sich durch polynomische Trendlinien (2. Ordnung) sehr gut annähern.

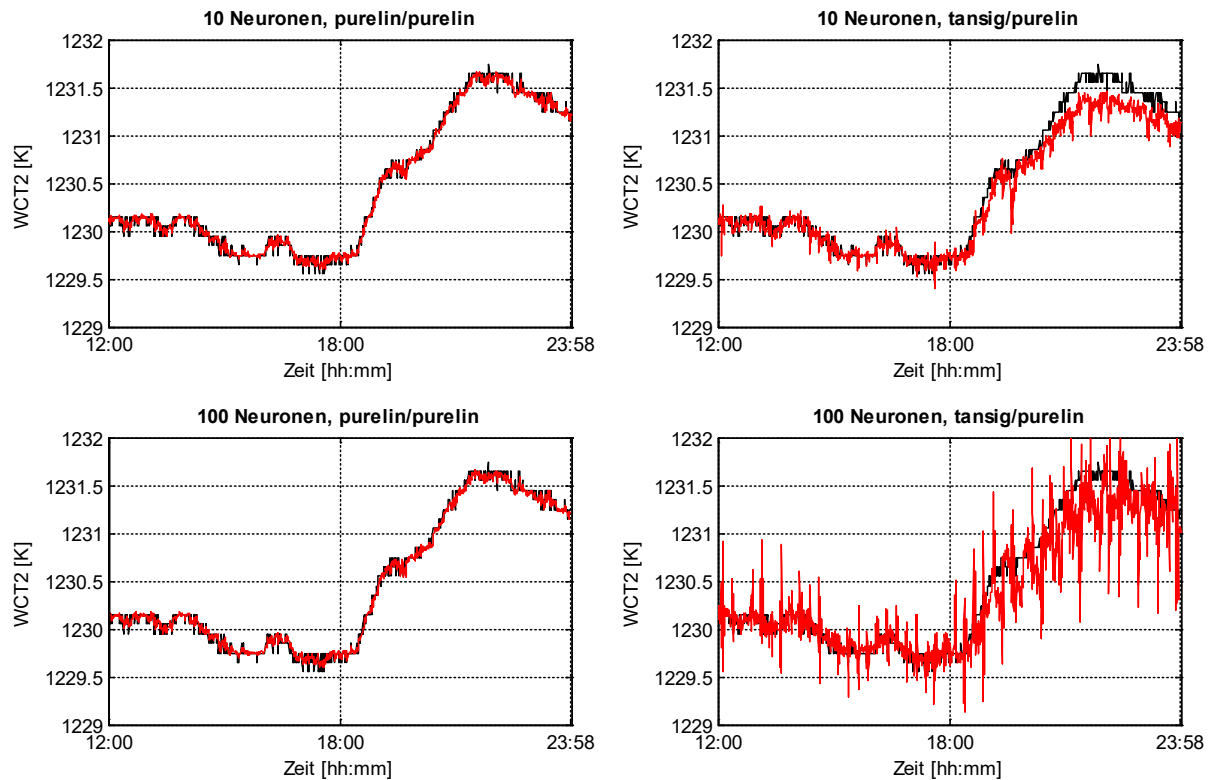


Abbildung 3.10: Vergleich der Testergebnisse für WCT2 bei verschiedenen Neuronenzahlen und Aktivierungsfunktionen

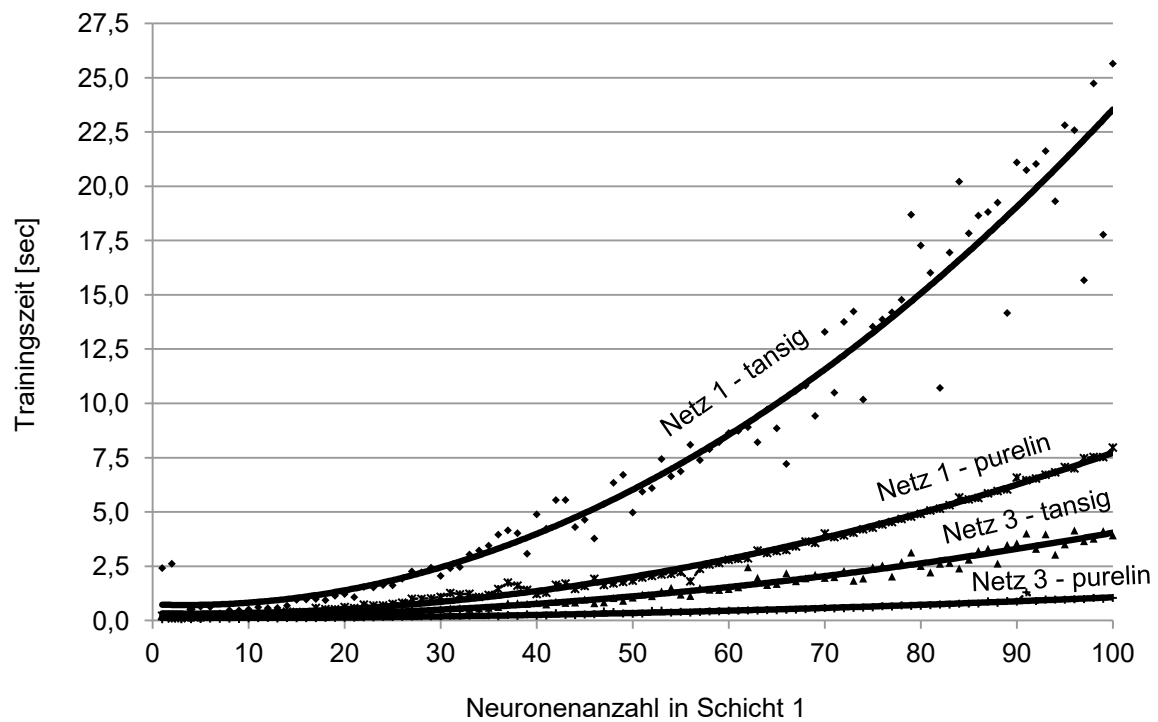


Abbildung 3.11: Trainingszeiten der Netze 1 und 3 in Abhängigkeit der Neuronenanzahl und der Aktivierungsfunktion

In Abbildung 3.11 ist erkennbar, dass die Trainingszeiten bei Netzen mit der sigmoiden Aktivierungsfunktion *tansig* steiler ansteigen. Zudem wird deutlich, dass Netz 1, welches in Summe mehr Ein- und Ausgangsgrößen enthält, ebenfalls einen erhöhten Rechenaufwand benötigt. Der polynomische Trend gilt für alle fünf Netze. Abbildung 3.12 bestätigt diesen Sachverhalt durch eine Normierung der benötigten Trainingszeiten auf ihren jeweiligen Wertebereich. Zur Darstellung dieses Diagramms wurde die lineare Aktivierungsfunktion verwendet. Mit der sigmoiden Funktion ergibt sich ein ähnliches Bild.

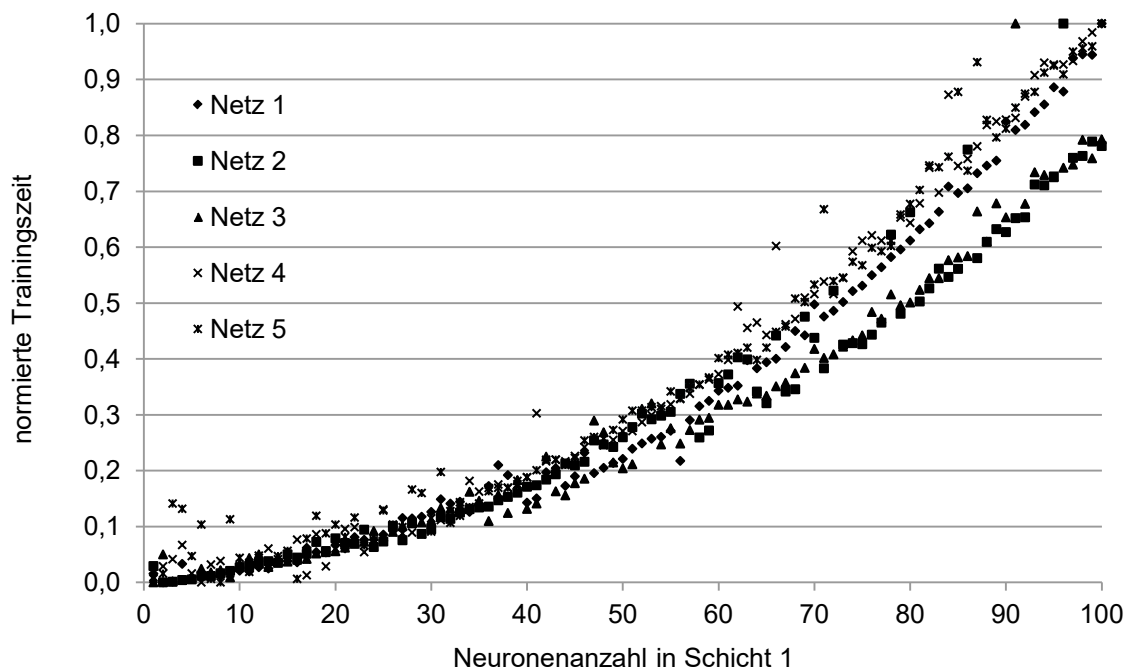


Abbildung 3.12: Normierte Trainingszeiten in Abhängigkeit der Neuronenanzahl

### 3.2.5 Einfluss der Tapped Delay Lines

Der TDL-Wert legt die Anzahl an zurückliegenden Messwerten pro Prozessgröße fest, die in die Berechnung des zukünftigen Ausgangswertes mit einfließen. Bei dieser Untersuchung wurden die TDL schrittweise bis 120 (= 60 min) vergrößert. Die Standardabweichungen sind in Abbildung 3.13 dargestellt. Bei den Netzen 1, 2, 3 und 5 führen steigende TDL-Werte entgegen den Erwartungen zu zunehmend schlechteren Ausgangssignalen. Als Ursache kann auch hier der steigende Rechenaufwand angenommen werden. Die Ergebnisse vom NIO-Netz 4 entsprechen den Erwartungen. Je mehr zurückliegende Werte der Gemengezugabe BATCH herangezogen werden, desto besser kann der Glasstand GLASS berechnet werden. Die Standardabweichungen vom NIO-Netz 4 verzeichnen ab TDL = 80 eine Sättigung. Es tritt keine sichtbare Verbesserung mehr ein.

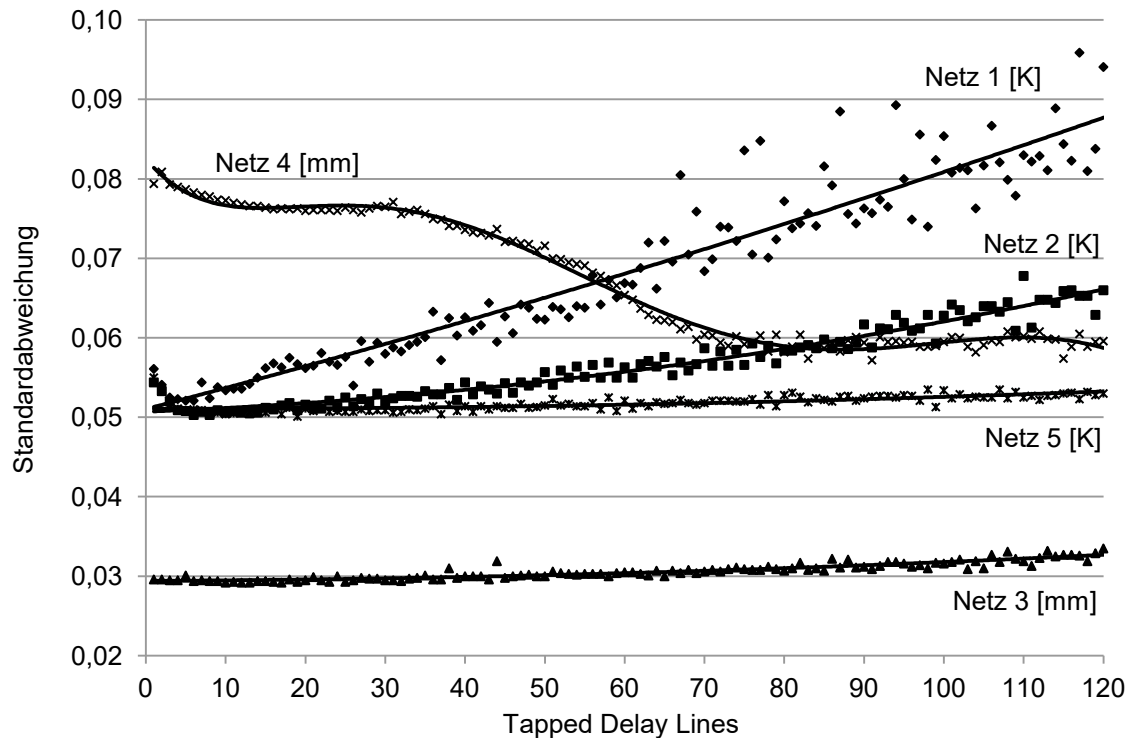


Abbildung 3.13: Standardabweichungen und Trendlinien in Abhängigkeit der TDL

Das NAR-Netz 5 zeigt den gleichen Trend wie die NARX-Netze, jedoch ist der Anstieg der Standardabweichungen über die gesamte TDL-Spanne deutlich geringer. Hier liegt die Differenz der Standardabweichungen zwischen 10 und 120 TDL bei minimalen 0,0026 K. Für Netz 1 beträgt diese Differenz 0,0407 K. Ein Unterschied der zeitlichen Verläufe von Netz 5 ist für verschiedene TDL-Werte kaum erkennbar.

Die Abbildung 3.14 veranschaulicht die bisherigen Erkenntnisse zum Einfluss der TDL und Aktivierungsfunktionen anhand einer direkten Gegenüberstellung der zeitlichen Verläufe für die Ausgangsgröße WCT1 (Netz 2) und GLASS (Netz 4). In den Diagrammen (a) und (b) sind die Testergebnisse für TDL = 10 und in (c) und (d) für TDL = 120 zu sehen. In jedem Diagramm sind von 12:00 - 18:00 Uhr die Testergebnisse mit linearer (rot) und ab 18:00 Uhr mit sigmoider (blau) Aktivierungsfunktion dargestellt. Bei Betrachtung der Simulationsergebnisse für Netz 2 fällt auf, dass für TDL = 120 der berechnete Verlauf zunehmenden Schwankungen unterworfen ist, die bei Verwendung der *tansig*-Aktivierungsfunktion ein nicht mehr akzeptables Maß erreichen. Lediglich bei Netz 4 kann mit einem höheren TDL-Wert eine Verbesserung erzielt werden, allerdings nur mit linearer Aktivierungsfunktion.

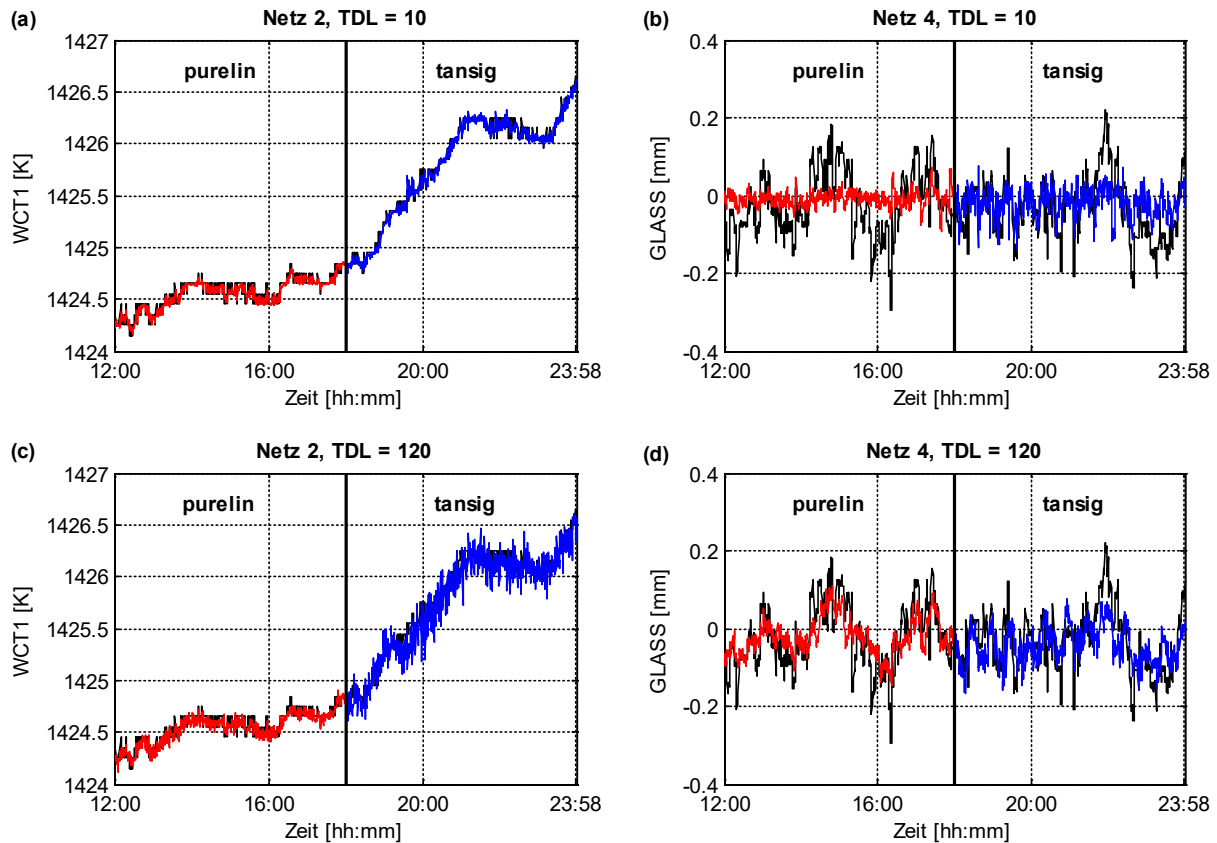


Abbildung 3.14: Zusammenfassende Gegenüberstellung der Testergebnisse von Netz 1 und 4 für verschiedene TDL und Aktivierungsfunktionen

Einen weiteren Nachweis zur Eignung der linearen Aktivierungsfunktion und zum Einfluss der Tapped Delay Lines liefert Tabelle 3.4, in der für alle fünf Netze die Standardabweichungen im Testzeitraum aufgelistet sind. Mit der von *MATLAB* empfohlenen Aktivierungsfunktion *tansig* lassen sich durchweg schlechtere Ergebnisse erzielen. Aus diesem Grund werden fortan alle weiterführenden Simulationen ausschließlich mit der linearen Aktivierungsfunktion *purelin* durchgeführt.

Tabelle 3.4: Vergleich der Standardabweichungen im Testzeitraum mit den Aktivierungsfunktionen *purelin* und *tansig* und den TDL-Werten 10 und 120

Netz	Aktivierungsfunktionen: <i>purelin</i> / <i>purelin</i>			Aktivierungsfunktionen: <i>tansig</i> / <i>purelin</i>		
	TDL = 10	TDL = 120	$\Delta$	TDL = 10	TDL = 120	$\Delta$
1	0,0534 K	0,0941 K	+76 %	0,1497 K	0,4452 K	+197 %
2	0,0506 K	0,0660 K	+30 %	0,0545 K	0,1165 K	+113 %
3	0,0292 mm	0,0335 mm	+14 %	0,0312 mm	0,0426 mm	+36 %
4	0,0773 mm	0,0596 mm	-23 %	0,0756 mm	0,0632 mm	-16 %
5	0,0504 K	0,0530 mm	+5 %	0,1548 K	0,2276 K	+47 %

Nachfolgend wird die Abhängigkeit der TDL-Werte in Bezug auf den benötigten Rechenaufwand betrachtet. Der TDL-Wert bestimmt die Dimension mehrerer Vektoren und Matrizen und hat demzufolge einen ähnlichen Einfluss wie die Anzahl der verborgenen Neuronen (vgl. Abbildung 3.12). Abbildung 3.15 verdeutlicht diesen Sachverhalt. Auch hier können die Verläufe durch eine polynomische Trendlinie zweiter Ordnung sehr gut dargestellt werden.

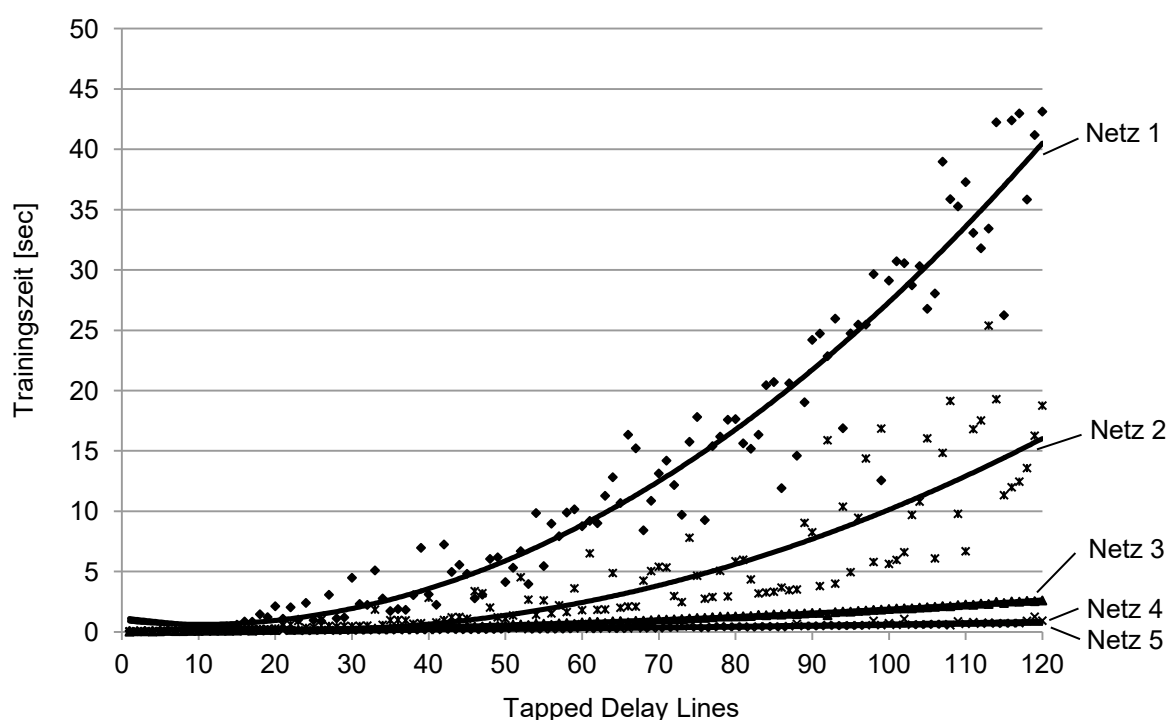


Abbildung 3.15: Trainingszeiten und Trendlinien in Abhängigkeit der TDL

Abschließend sind die zusammenhängenden Wertebereiche der Tapped Delay Lines aufgelistet, mit denen die höchste Genauigkeit erzielt werden kann. Entgegen den Erwartungen führen bei vier von fünf Netzen weniger zurückliegende Messdaten zu einem besseren Ergebnis. Lediglich das NIO-Netz 4 benötigt einen ausreichend großen zurückliegenden Zeithorizont.

	<u>geeigneter Wertebereich</u>	<u>Wahl</u>
▪ Netz 1:	$3 \leq \text{TDL} \leq 12$	TDL = 10
▪ Netz 2:	$4 \leq \text{TDL} \leq 14$	TDL = 10
▪ Netz 3:	$6 \leq \text{TDL} \leq 19$	TDL = 10
▪ Netz 4:	$78 \leq \text{TDL} \leq 120$	TDL = 120
▪ Netz 5:	$8 \leq \text{TDL} \leq 32$	TDL = 10



### 3.2.6 Einfluss der Trainings- und Testzeiträume

Der Umfang nutzbarer Informationen wurde bisher durch Änderungen an der Netzstruktur bestimmt. Eine weitere Möglichkeit den Informationsgehalt zu variieren, besteht in der Auswahl der Trainings- und Testzeiträume. Für diese Untersuchung wurde jedes Netz mit den Daten von stündlich verlängerten Trainingszeiträumen (1 - 12 Stunden) trainiert. Das Ende des Trainingszeitraums ist der Beginn des Testzeitraums, der ebenfalls aufsteigend von 1 - 12 Stunden variiert wurde. Die Auswertung der Simulationsergebnisse wird allein auf die Auswertung der Standardabweichungen reduziert. Die maximalen Fehler und Trainingszeiten sind nicht aussagekräftig. Liegt in der ersten Stunde im Testzeitraum ein betragsmäßig großer Fehler vor, ist dieser zumeist auch in den folgenden Stunden der größte Fehler. Zudem hängen die Ergebnisse in einem unbestimmten Maß vom originalen zeitlichen Verlauf der Ausgangsgröße ab, da unterschiedliche Trainings- und Testzeiträume betrachtet wurden. Die berechneten Standardabweichungen sind gegenüber Ausreißern und dem tatsächlichen Verlauf der Ausgangsgröße robust. Tabelle 3.5 listet die durchschnittlichen Standardabweichungen (S) für ausgewählte Trainings- und Testzeiten auf.

Tabelle 3.5: Durchschnittliche Standardabweichungen in Abhängigkeit der Trainings- und Testdauer

		Durchschnittliche Standardabweichungen				
		Netz 1	Netz 2	Netz 3	Netz 4	Netz 5
Trainingsdauer	1 h	0,0850 K	0,0675 K	0,0342 mm	-	0,0544 K
	3 h	0,0534 K	0,0534 K	0,0297 mm	0,0761 mm	0,0485 K
	6 h	0,0551 K	0,0528 K	0,0301 mm	0,0569 mm	0,0536 K
	9 h	0,0518 K	0,0519 K	0,0281 mm	0,0595 mm	0,0499 K
	12 h	0,0509 K	0,0524 K	0,0291 mm	0,0550 mm	0,0488 K
Testdauer	1 h	0,0543 K	0,0519 K	0,0302 mm	0,0545 mm	0,0492 K
	3 h	0,0554 K	0,0539 K	0,0303 mm	0,0613 mm	0,0500 K
	6 h	0,0562 K	0,0545 K	0,0302 mm	0,0653 mm	0,0505 K
	9 h	0,0566 K	0,0543 K	0,0299 mm	0,0662 mm	0,0510 K
	12 h	0,0574 K	0,0540 K	0,0299 mm	0,0658 mm	0,0511 K

Zum besseren Verständnis wird nachfolgend die Berechnung der ersten Standardabweichung (0,0850 K) erläutert. Das Netz wurde mit den Daten der ersten Stunde trainiert. Der Testzeitraum beginnt unmittelbar nach dem Training (01:00 Uhr) und wird schrittweise um eine Stunde, bis maximal zwölf Stunden (01:00 - 13:00 Uhr), verlängert. Der dabei

ermittelte Durchschnitt aller Standardabweichungen ist in der Tabelle angegeben. Für Netz 4 ist das Training mit den Daten der ersten Stunde nicht möglich, da dieses Netz mit  $TDL = 120$  konfiguriert ist.

Ein Vergleich der Werte zeigt, dass die Fehler bei einem einstündigen Trainingszeitraum am größten sind. Mit längerem Trainingszeitraum nehmen die Standardabweichungen weiter ab, jedoch in zunehmend geringerem Maß. Dieser Trend lässt sich für alle Netze vereinfacht auf einen hyperbelförmigen Verlauf reduzieren (Abbildung 3.16, (a)).

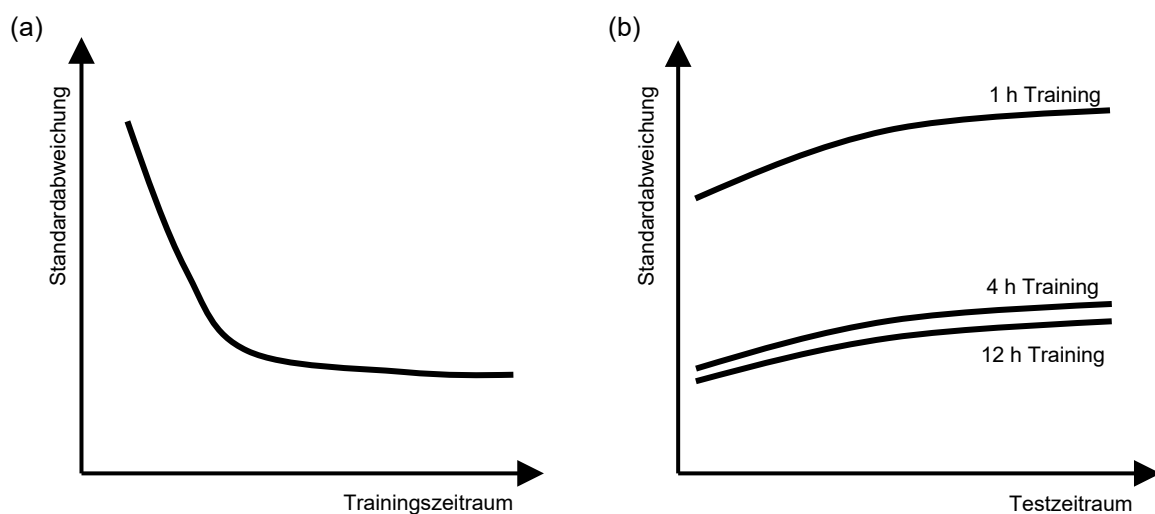


Abbildung 3.16: Allgemeine Tendenz der Standardabweichungen in Abhängigkeit der Länge vom Trainings- (a) und Testzeitraum (b)

Die Anzahl an Trainingsstunden, ab der die Ergebnisse sich nur noch geringfügig verbessern, ist von Netz zu Netz unterschiedlich. Als allgemeiner Richtwert kann ein Trainingszeitraum von mindestens vier Stunden angesetzt werden. Für alle fünf Netze beträgt die durchschnittliche prozentuale Verbesserung der Standardabweichungen innerhalb der ersten drei Trainingsstunden 21 %. Zwischen vier und zwölf Stunden Training verringern sich die Standardabweichungen lediglich um weitere 5 %. Der vereinfachte Verlauf der Standardabweichungen in Abhängigkeit der Testdauer (Abbildung 3.16, (b)) veranschaulicht diesen Sachverhalt. Auch hier ist eine Sättigung erkennbar.

Die benötigte Rechenzeit zum Training der KNN liegt deutlich unter der Schrittweite des Datensatzes, weshalb längere Trainingszeiträume für die fünf ausgewählten Netze ohne Bedenken gewählt werden können.

### 3.2.7 Reproduzierbarkeit der Simulationsergebnisse

Die Reproduzierbarkeit sämtlicher Simulationsergebnisse wird durch die übliche zufällige Initialisierung der Wichtungsmatrizen und Schwellwertvektoren beim Training gefährdet. Dieser Sachverhalt zeigt sich darin, dass KNN bei gleicher Konfiguration und gleicher Datenbasis nach erneutem Training nicht die exakt gleichen Testergebnisse liefern. Diese Streuung nimmt mit größerem Rechenaufwand zu. Für eine vollständige Reproduzierbarkeit ist eine Modifizierung des offenen *MATLAB*-Quellcodes notwendig.

#### 3.2.7.1 Analyse der Reproduzierbarkeit bei zufälliger Initialisierung

Vor dem ersten Iterationsschritt werden die Wichtungsmatrizen und Schwellwertvektoren mit Zufallswerten im Bereich  $[-1 \ 1]$  initialisiert. Diese Werte definieren den Ausgangspunkt im Optimierungsalgorithmus. Nach jedem Iterationsschritt werden die Gewichte und Schwellwerte angepasst. Die Iteration wird bis zum Auffinden eines lokalen oder globalen Minimums fortgesetzt, sofern keine Abbruchbedingung erfüllt wird.

Zur Bewertung der Reproduzierbarkeit bei zufälliger Initialisierung wurden vier verschiedene Parameterkombinationen getestet (Tabelle 3.6). Ausgehend von der bisher für jedes Netz evaluierten Konfiguration (1) sind die gezielten Änderungen (2) - (4) vorgenommen worden.

Tabelle 3.6: Untersuchte Netzkonfigurationen zur Bewertung der Reproduzierbarkeit

			Netz 1	Netz 2	Netz 3	Netz 4	Netz 5	
Konfiguration	(1)	Typ	NARX	NARX	NARX	NIO	NAR	
		Eingänge	BATCH	GFC	BATCH	BATCH		
			GFC	DAF				
			CAF					
			DAF					
		Ausgang	WCT2	WCT1	GLASS	GLASS	WCT2	
		Aktivierungsfunktionen	purelin / purelin					
		Neuronenanzahl	10					
		Tapped Delay Lines	10	10	10	120	10	
	(2)	Neuronenanzahl	100					
	(3)	Tapped Delay Lines	120	120	120	10	120	
	(4)	Aktivierungsfunktionen	tansig / purelin					

Die Netze wurden zehnmal hintereinander trainiert und getestet. Zur Bewertung dient zum einen die gleichgerichtete durchschnittliche Änderung der Standardabweichungen vom Mittelwert aller zehn Durchläufe und zum anderen die maximale Spannweite der berechneten Fehler. Letztere berechnet sich aus zwei Extremwerten und berücksichtigt somit eventuelle Ausreißer. Abbildung 3.17 zeigt für Netz 1 mit der Konfiguration (3) die berechneten Standardabweichungen (a) und Spannweiten (b) für alle zehn Simulationsdurchläufe und die daraus ermittelten Kennzahlen zur Bewertung der Reproduzierbarkeit.

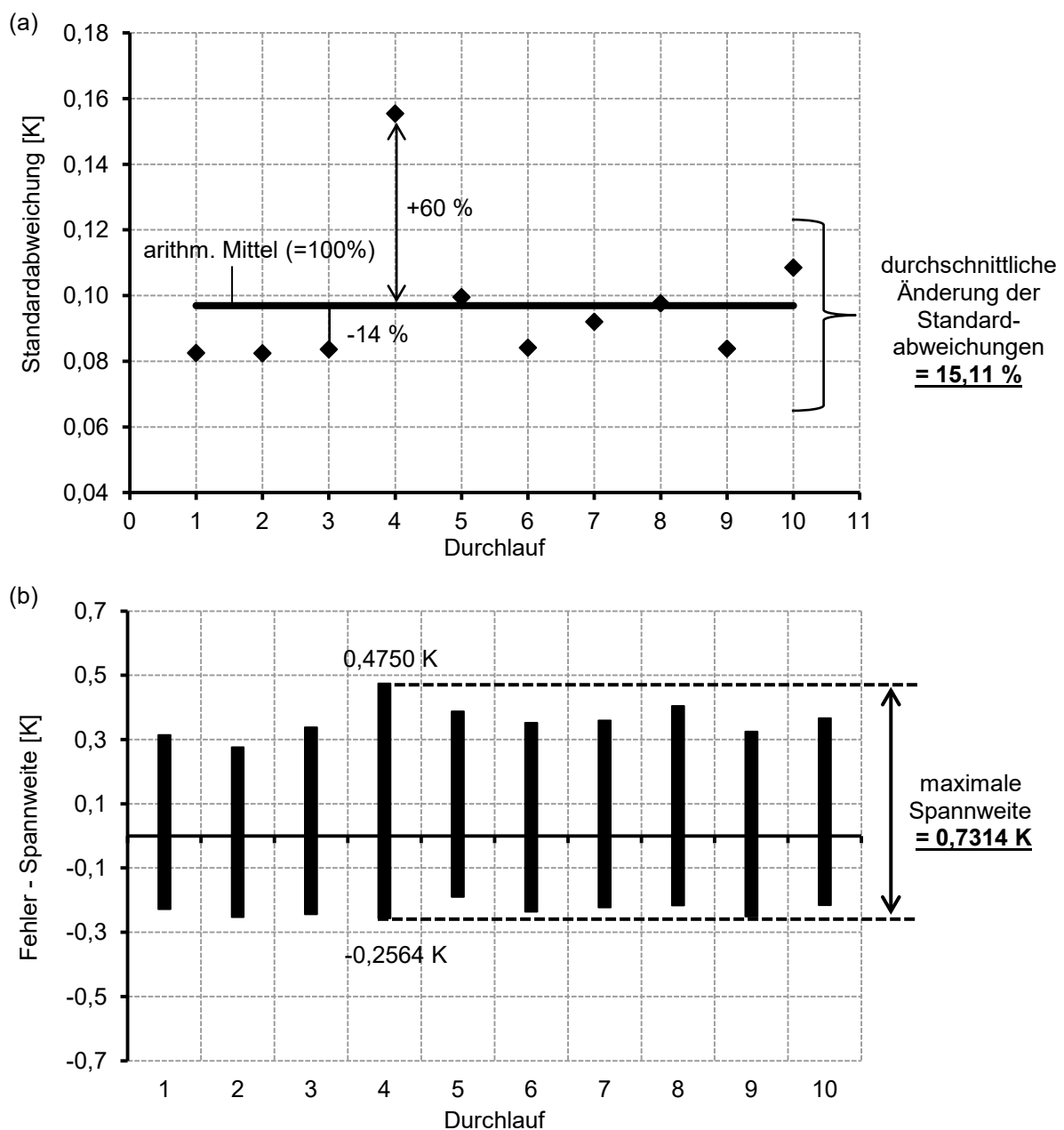


Abbildung 3.17: Standardabweichungen (a) und Spannweiten (b) von Netz 1 mit 120 TDL bei zehn Simulationsdurchläufen

Die Diagramme zeigen, dass bei mehrmaligem Training mit  $TDL = 120$  die Standardabweichungen von Netz 1 um ca. 15 % schwanken. Die maximale Spannweite der Fehler beträgt für diese Konfiguration 0,7314 K. Beide Werte sind vergleichsweise hoch und sprechen für eine geringe Reproduzierbarkeit. Diese beiden Kennzahlen sind in Abbildung 3.18 für alle Netze und alle Parameterkonfigurationen in einem gemeinsamen Streudiagramm aufgetragen. Hinter jedem Datenpunkt ist das dazugehörige Netz angegeben. Im linken Diagramm ist zu erkennen, dass die Netze mit den Aktivierungsfunktionen *tansig* / *purelin* (Konfig. 4) eine beträchtliche Streuung aufweisen. Die Standardabweichungen von Netz 5 schwanken um 29,5 %, bei Netz 1 um 18,5 %. Eine maximale Spannweite von über 2 K deutet bei Netz 1 zusätzlich auf vermehrte Ausreißer hin. Die Untersuchung bestätigt damit die Erkenntnisse zum Einfluss der Aktivierungsfunktion *tansig* (vgl. Abschnitte 3.2.3 und 3.2.4).

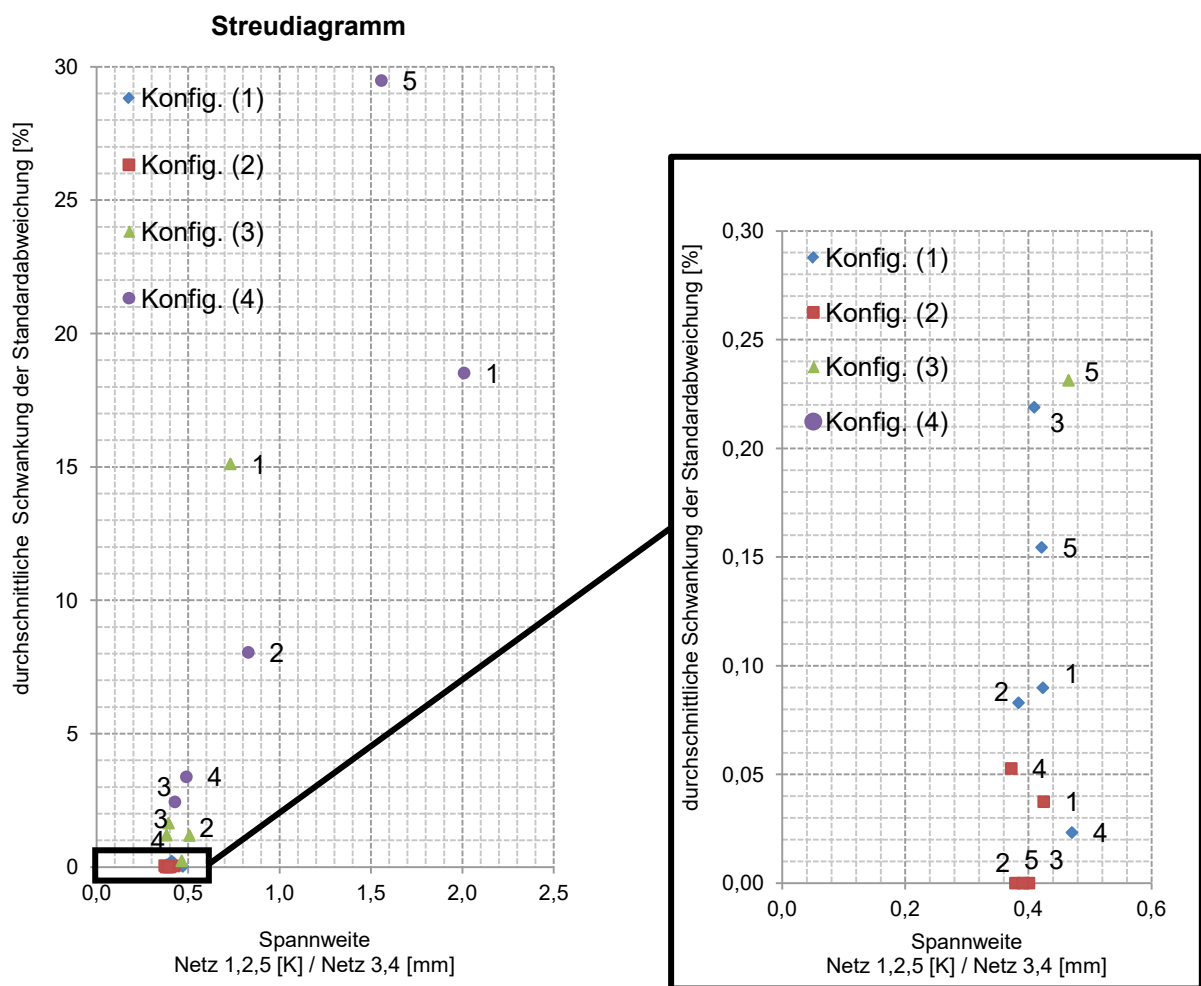


Abbildung 3.18: Streudiagramm zur Bewertung der Reproduzierbarkeit bei Änderung ausgewählter Netzparameter (Bem.: Konfigurationen laut Tabelle 3.6, die Nummer des Netzes ist am Datenpunkt angegeben)

Im rechten Diagrammausschnitt sind die Netze zu sehen, deren Standardabweichungen und Spannweiten sich in allen Simulationsdurchläufen nur geringfügig ändern. Die größte durchschnittliche Änderung der Standardabweichungen von den Netzen mit der optimalen Konfiguration (1) weist Netz 3 mit rund 0,22 % auf. Die höchste Reproduzierbarkeit erzielten die Netze mit der Konfiguration (2). Das bedeutet, dass eine höhere Anzahl an Neuronen die Dispersion der berechneten Ausgangsgrößen verringert, wenngleich sich die benötigte Trainingszeit verlängert (vgl. Abschnitt 3.2.4).

#### 3.2.7.2 Modifizierung der Trainingsumgebung

Für eine vollständige Reproduzierbarkeit muss die von *MATLAB* bereitgestellte Trainingsfunktion `train` modifiziert werden. Diese Funktion benötigt als Argumente das zu trainierende Netz, die tatsächlichen Ein- und Ausgangsgrößen im Trainingszeitraum und die vor dem Trainingszeitraum zurückliegenden Messwerte (States). Nach dem Training gibt *MATLAB* das trainierte Netz *net* und einen Trainingsbericht (*train record*, *tr*) aus.

```
[net, tr] =  
    train(net, TrainInputs, TrainTargets, TrainInputStates, TrainLayerStates)
```

Der Trainingsalgorithmus wird im Objekt *net* definiert. Die tiefgreifende objektorientierte Struktur des *MATLAB*-Codes erhöht das Risiko einer fehlerhaften Programmierung. Aus diesem Grund ist lediglich ein zusätzlicher Schritt im Trainingsalgorithmus eingeführt worden. Nach der Dimensionierung und zufälligen Initialisierung der Schwellwerte *b* und Wichtungsmatrizen *IW* und *LW* werden alle Werte mit einem konstanten Wert (hier: 1) überschrieben. Somit liefert jedes Netz nach erneutem Training exakt die gleichen Ergebnisse. Die Befehle sind nachfolgend angegeben.

```
net.b{1,1}(:, :) = 1;  
net.IW{1,1}(:, :) = 1;  
net.LW{2,1}(:, :) = 1;
```

#### 3.2.7.3 Verifikation der bisherigen Simulationsergebnisse

Die Änderung des *MATLAB*-Quellcodes stellt einen direkten Eingriff in das Training dar. Zur Verifikation der bisherigen Simulationsergebnisse stellt Abbildung 3.19 für die Netze 1 und 4 die berechneten Standardabweichungen und benötigten Trainingszeiten in Abhängigkeit der TDL für beide Initialisierungsmethoden gegenüber.

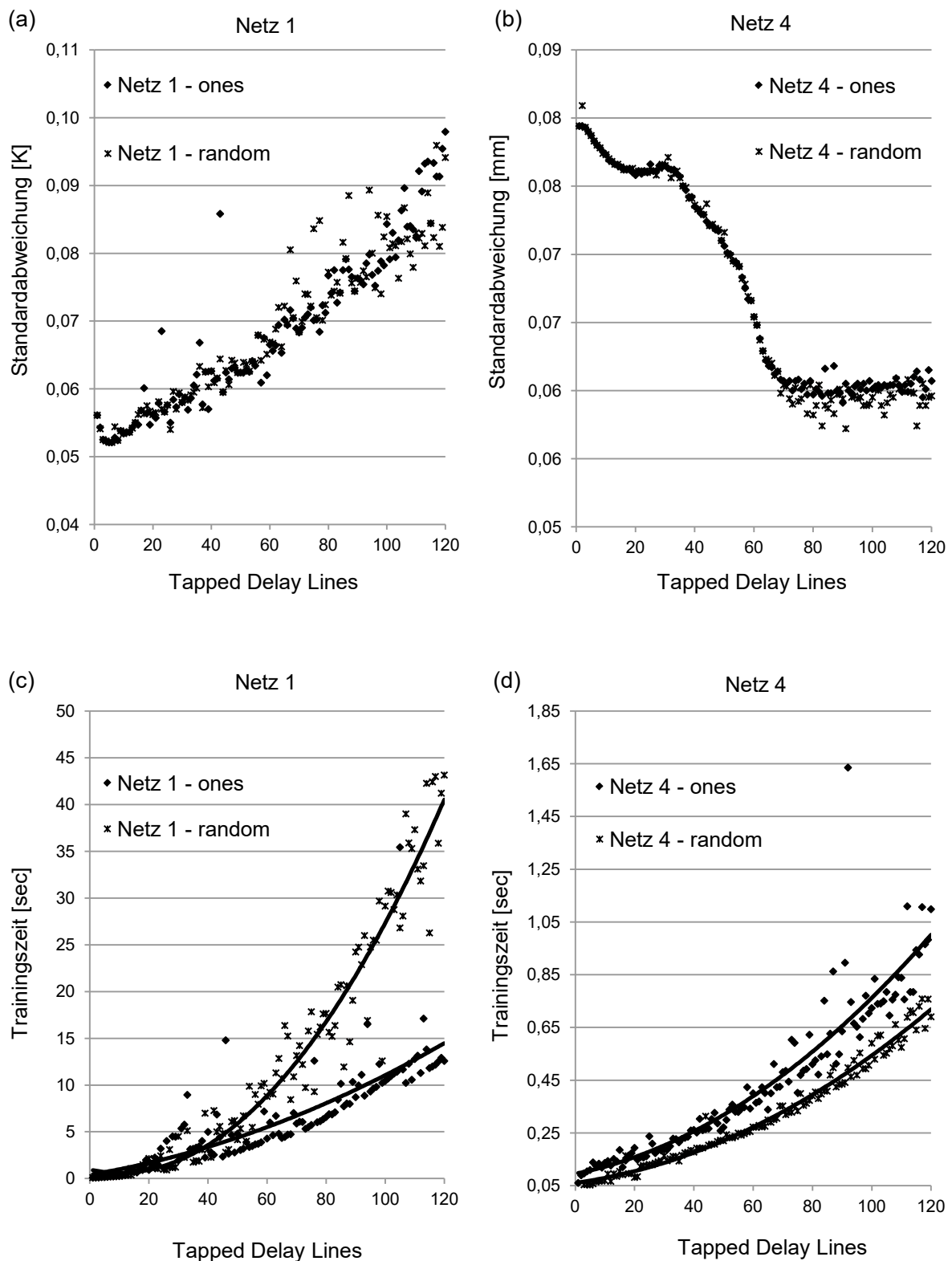


Abbildung 3.19: Gegenüberstellung der berechneten Standardabweichungen (a), (b) und benötigten Trainingszeiten (c), (d) der Netze 1 und 4 mit konstanter (ones) und zufälliger (random) Initialisierung

In den Diagrammen (a) und (b) ist erkennbar, dass sich der Einfluss der TDL auf die Standardabweichungen nicht geändert hat. Die Schwankungsbreite bei höheren TDL-Werten nimmt jedoch geringfügig zu. Dies gilt gleichermaßen für die anderen Netze. Die benötigten Trainingszeiten haben sich allerdings geändert. Diese sind von Netz zu Netz unterschiedlich, dennoch kann für alle fünf Netze erneut eine polynomische Trendlinie angenähert werden. Der Einfluss der verborgenen Neuronen und der Aktivierungsfunktion *tansig*, in den Diagrammen nicht dargestellt, hat sich durch die konstante Initialisierung nicht geändert. Dies wurde durch stichprobenartige Simulationen überprüft.

Es ist aufgefallen, dass bei rechenintensiven Netzkonfigurationen die Resultate bei minimalen Änderungen der Netzstruktur mitunter erheblichen sprunghaften Änderungen unterliegen können. Die Standardabweichung von Netz 1 beträgt z. B. für  $TDL = 99$  noch 0,0782 K, für  $TDL = 100$  bereits 0,0843 K. Das entspricht, bezogen auf den gesamten Wertebereich, einer prozentualen Änderung von 13,3 %. Die Ursache für diese zunehmenden Schwankungen liegt allein am Rechenaufwand. Die Anzahl an Multiplikationen steigt mit der Dimension der Wichtungsmatrizen. Eine Analyse der Trainingsberichte hat gezeigt, dass die Optima in einem sehr flachen Bereich liegen (Gradienten von  $10^{-7}$  bis  $10^{-11}$ ). Multiplikationen von kleinen Zahlen führen zu noch kleineren Zahlen, die bis an die Grenze des Genauigkeitsbereichs reichen. Die Maschinengenauigkeit beträgt in *MATLAB*  $2,2204 \cdot 10^{-16}$ . Die Vielzahl an Multiplikationen mit sehr kleinen Zahlen führt daher zwangsläufig zu vermehrten numerischen Ungenauigkeiten und kumulierten Rundungsfehlern.

#### 3.2.8 Vergleich der Simulationsergebnisse mit dem vereinfachten Datensatz

Die Aufbereitung des vereinfachten Datensatzes ist im Abschnitt 2.2.3 (Abbildung 2.14) beschrieben. Es wurden die gleichen Untersuchungen mit diesem Datensatz durchgeführt. Die Ergebnisse werden in diesem Abschnitt kurz zusammengefasst.

Der Einfluss der Neuronen und der Tapped Delay Lines wird zusammenfassend in Abbildung 3.20 veranschaulicht. Tabelle 3.7 listet zusätzlich die Standardabweichungen und Trainingszeiten auf. Die Ergebnisse fallen sehr unterschiedlich aus. Eine Erhöhung der TDL-Werte (b) oder Neuronen (c) führt bei Netz 1 zu stark verrauschten Ausgangswerten, während bei Netz 3 (d) keine sichtbaren Veränderungen erkennbar sind. Bei geeigneter Konfiguration sind die Verläufe nahezu deckungsgleich. Die Standardabweichung von Netz 3 beträgt mit  $S^1 = TDL = 10$  minimale 0,00013 mm. Die Standardabweichung ist somit 300-mal geringer als bei Verwendung des ursprünglichen Datensatzes.



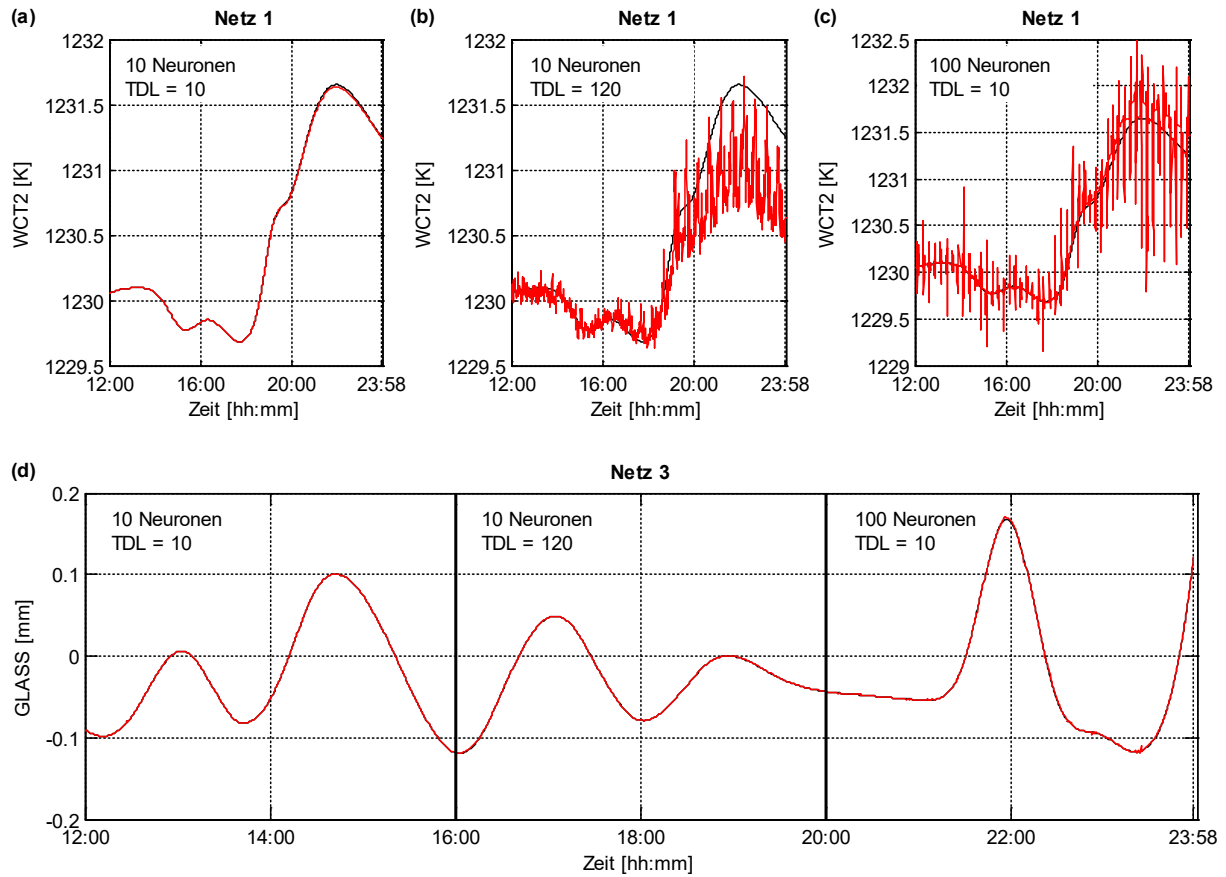


Abbildung 3.20: Testergebnisse der Netze 1 und 3 für verschiedene Konfigurationen bei Verwendung des vereinfachten Datensatzes

Tabelle 3.7: Vergleich der Standardabweichungen  $S$  und Trainingszeiten  $T_{train}$  bei verschiedenen Konfigurationen der TDL und Neuronen

	Konfiguration			Datensatz	
	TDL	$S^1$		original	vereinfacht
Netz 1	10	10	$S$	0,1033 K	0,0066 K
			$T_{train}$	0,52 sec	41,99 sec
	120	10	$S$	0,4452 K	0,2998 K
			$T_{train}$	21,39 sec	41,51 sec
	10	100	$S$	0,2714 K	0,2171 K
			$T_{train}$	25,65 sec	1.241,70 sec
Netz 3	10	10	$S$	0,0365 mm	0,0001 mm
			$T_{train}$	0,20 sec	0,99 sec
	120	10	$S$	0,0426 mm	0,0007 mm
			$T_{train}$	3,14 sec	47,85 sec
	10	100	$S$	0,0376 mm	0,0009 mm
			$T_{train}$	3,93 sec	58,09 sec

Die durchgeführten Mittelwertbildungen und Interpolationen haben die zeitlichen Verläufe der tatsächlich gemessenen Prozessgrößen stark geglättet. Der Informationsgehalt des Datensatzes und damit das zu lernende Prozessverhalten haben sich wesentlich vereinfacht, was letztlich zu diesen scheinbar guten Ergebnissen führt. Eine kritische Betrachtung ist dennoch zwingend notwendig. Vereinzelt sind nicht plausible Ergebnisse und ungedämpfte Schwingungen aufgetreten, die darauf hindeuten, dass die zu verarbeitenden Informationen nicht mehr zusammenpassen. Die benötigten Trainingszeiten haben sich um ein Vielfaches verlängert, wodurch die Tauglichkeit des Datensatzes zusätzlich hinterfragt werden muss. Die erheblich längeren Simulationszeiten resultieren aus der vermehrten Anzahl an Iterationen, was ebenfalls den mangelnden Informationsgehalt des Datensatzes untermauert.

Der Einfluss der Neuronen, der Tapped Delay Lines, der Trainings- und Testzeiträume ist im Vergleich zum originalen Datensatz prinzipiell gleich. Für eine Regelung sind die mitunter minutenlangen Rechenzeiten allerdings nicht akzeptabel. Der Aufwand zur Aufbereitung der hinzukommenden Messdaten und deren Integration in den vom Netz benötigten Datenhorizont darf ebenfalls nicht vernachlässigt werden.

### 3.3 Zusammenfassung

Die durchgeführten Simulationen haben gezeigt, dass sich Künstliche Neuronale Netze zur Berechnung von Prozessgrößen in Flachglasschmelzwannen eignen. Auf Basis des vorliegenden Datensatzes konnten insgesamt fünf Netze modelliert werden, mit denen u. a. die qualitätsbestimmende Decktemperatur WCT2 und der Glasstand GLASS hinreichend genau berechnet werden können. In [48] wurde der gewünschte Genauigkeitsbereich der konventionellen Temperaturregelungen in Glasschmelzwannen auf  $\pm 1$  K beziffert. Mit den evaluierten KNN ist es möglich, die relevanten Prozessgrößen deutlich genauer zu berechnen, womit diese Modelle für eine prädiktive Regelung prädestiniert sind. In der Tabelle 3.8 sind für alle Netze die optimalen Konfigurationen und die dazugehörigen statistischen Kennzahlen der berechneten Fehler im Testzeitraum zusammengefasst.

Von den verschiedenen Netztypen erzielen NAR-Netze die höchste Genauigkeit. Aufgrund der fehlenden Eingangsgrößen ist dieser Netztyp für Regelzwecke allerdings nicht geeignet. Vor dem Hintergrund, dass das NIO-Netz 4 den Glasstand GLASS lediglich anhand der Mengenzufuhr BATCH berechnet, sind die erzielten Ergebnisse erstaunlich gut. Damit bestätigt sich der erwartet direkte Zusammenhang dieser beiden Größen.

Tabelle 3.8: Übersicht der evaluierten Netzkonfigurationen und Testergebnisse

	Netz 1	Netz 2	Netz 3	Netz 4	Netz 5
Typ	NARX	NARX	NARX	NIO	NAR
Modus	open loop (One Step Prediction)				
Eingangsgrößen	BATCH	GFC	BATCH	BATCH	
	GFC	DAF			
	CAF				
	DAF				
Ausgangsgrößen	WCT2	WCT1	GLASS	GLASS	WCT2
Trainingsmethode	Levenberg-Marquardt ( <i>trainlm</i> )				
Aktivierungsfunktionen	<i>purelin</i> / <i>purelin</i>				
Neuronenanzahl Schicht 1	10				
Tapped Delay Lines	10	10	10	120	10
Trainingszeitraum	> 4 Stunden (Empfehlung 12 Stunden)				
Initialisierung	konstant, alle Werte auf 1				
Simulationsergebnisse (Testzeitraum 12 - 24 Uhr)					
max. pos. Fehler	0,2163 K	0,1891 K	0,2012 mm	0,1792 mm	0,1683 K
max. neg. Fehler	-0,2054 K	-0,1998 K	-0,1995 mm	-0,1927 mm	-0,2163 K
arithm. Fehlermittelwert	0,0074 K	0,0057 K	-0,000056 mm	-0,0016 mm	-0,0088 K
Standardabw. der Fehler	0,0534 K	0,0512 K	0,0292 mm	0,0607 mm	0,0506 K

Der Levenberg-Marquardt-Algorithmus benötigte die kürzeste Rechenzeit und die wenigsten Iterationen zum Training der Netze. Als Aktivierungsfunktion wird in beiden Schichten die lineare Funktion *purelin* empfohlen. Die für NARX-Netze von *MATLAB* standardmäßig gewählte Aktivierungsfunktion *tansig* führt bei zunehmendem Rechenaufwand zu einer sichtbar schlechteren Qualität der Ausgangssignale. Entgegen den Erwartungen führt eine Berücksichtigung von mehr zurückliegenden Messdaten mit Ausnahme von Netz 4 zu keiner Verbesserung. Der Trainingszeitraum sollte bei allen Netzen mindestens vier Stunden lang sein. Aufgrund der begrenzten Fließkommarithmetik sind rechenintensive Konfigurationen zu vermeiden. Für eine vollständige Reproduzierbarkeit der Simulationsergebnisse ist der Quellcode der *MATLAB*-Funktion `train` auf eine konstante Initialisierung anzupassen.

Die in diesem Kapitel beschriebenen Ergebnisse sind auf die One Step Prediction bezogen. Der hohe Genauigkeitsbereich aller Netze für die OSP ist beachtlich. Im folgenden Kapitel werden die evaluierten Netze durch eine gezielte Änderung der Netzstruktur zur Multi Step Prediction herangezogen und analog der in diesem Kapitel durchgeführten Untersuchungen der Einfluss der Netzparameter analysiert.

---

## 4 Multi Step Prediction

Bei der Multi Step Prediction verwendet das Netz anstatt der tatsächlichen die selbst berechneten Werte der Ausgangsgrößen. Nicht alle drei Netztypen sind für die MSP geeignet. Netze vom Typ NIO benötigen keine zurückgeführten Ausgangsgrößen, weshalb eine MSP mit diesem Netztyp nicht möglich ist. Diese Netze sind für eine modellbasierte prädiktive Regelstrategie dennoch geeignet, da sie ähnlich wie eine konventionelle Übertragungsfunktion rekursiv zur Prädiktion genutzt werden können. NAR-Netze verwenden ausschließlich Werte der Ausgangsgrößen. Damit ist eine Prädiktion über einen längeren Zeitraum fehleranfällig. Netze vom Typ NARX sind für eine MSP gut geeignet. Allerdings ziehen diese Netze im Prädiktionszeitraum weiterhin die original gemessenen Eingangsgrößen heran, was bei der Auswertung der Ergebnisse berücksichtigt werden muss.

Der Einfluss der Netzparameter wird systematisch analysiert. Im ersten Abschnitt wird die Simulationsumgebung beschrieben. Die Auswertung der Simulationsergebnisse erfolgt im zweiten Abschnitt. Das Kapitel schließt im dritten Abschnitt mit einer Zusammenfassung.

### 4.1 Simulationsumgebung und Methodik

Jedes Netz wurde open loop trainiert und anschließend in die "closed loop"-Architektur überführt. Abbildung 4.1 verdeutlicht noch einmal den Unterschied zwischen der OSP und MSP beispielhaft für das NARX-Netz 1 und das NAR-Netz 5.

Ausgangspunkt der Untersuchungen sind die evaluierten Konfigurationen zur OSP (Tabelle 3.8). Die einzelnen Netzparameter wurden systematisch verändert. Der Prädiktionszeitraum beginnt unmittelbar nach dem Ende des Trainingszeitraums. Die Prädiktionsdauer beträgt, wenn nicht anders angegeben, drei Stunden.

In einer weiteren Simulation (Abschnitt 4.2.3) wurden die Netze zwischen 12:00 und 21:00 Uhr stündlich neu trainiert. Dabei sind verschiedene Trainings- und Prädiktionshorizonte untersucht worden. Diese "gleitende" Testsimulation dient zur Bewertung der Praxistauglichkeit, wenn die KNN in regelmäßigen Abständen neu trainiert werden.

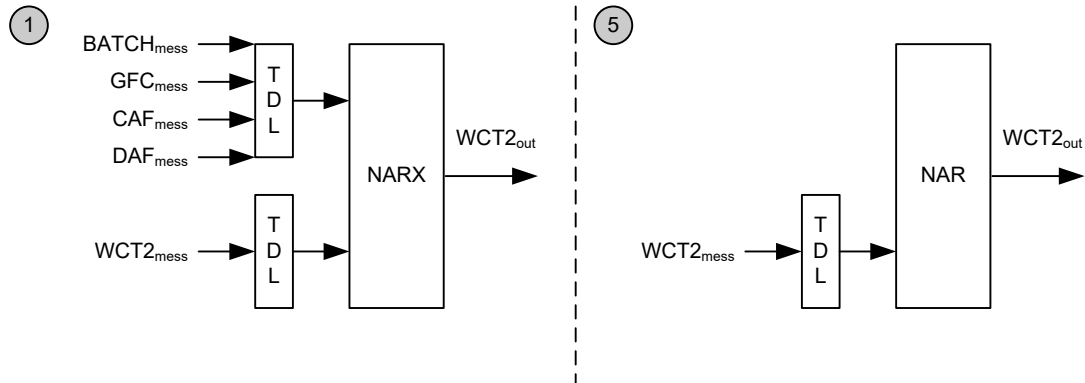
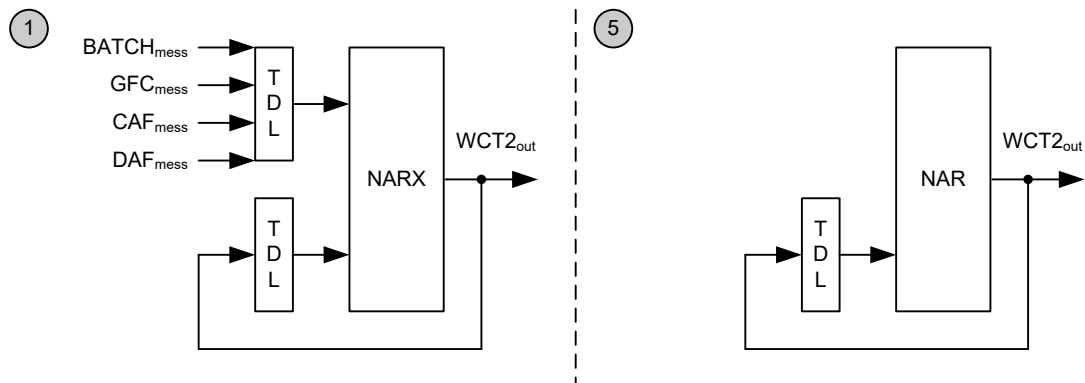
**Open Loop Architektur (One Step Prediction)****Closed Loop Architektur (Multi Step Prediction)**

Abbildung 4.1: Unterschied zwischen OSP und MSP am Beispiel der Netze 1 und 5

## 4.2 Simulationsergebnisse

### 4.2.1 Einfluss der verborgenen Neuronen

Die Prädiktionsergebnisse sind von Netz zu Netz unterschiedlich (Abbildung 4.2). Lediglich Netz 1 ist in der Lage, den Verlauf der tatsächlichen Temperatur WCT2 zumindest im Trend zu folgen. Die Standardabweichung beträgt im dargestellten Prädiktionszeitraum 0,0731 K, der maximale Fehler 0,1842 K. Das Signalverhalten der Zielgröße kann nicht nachgebildet werden. Dieser Sachverhalt gilt für alle Netze. Die Ergebnisse von Netz 2, 3 und 5 sind durchweg nicht zufriedenstellend.

Die Anzahl der Neuronen hat zusammenfassend keinen erkennbaren Einfluss auf die MSP. Bei Netz 2 und 5 treten vereinzelt Ausreißer auf. Die dargestellten Ergebnisse verdeutlichen den signifikanten Unterschied zur OSP.

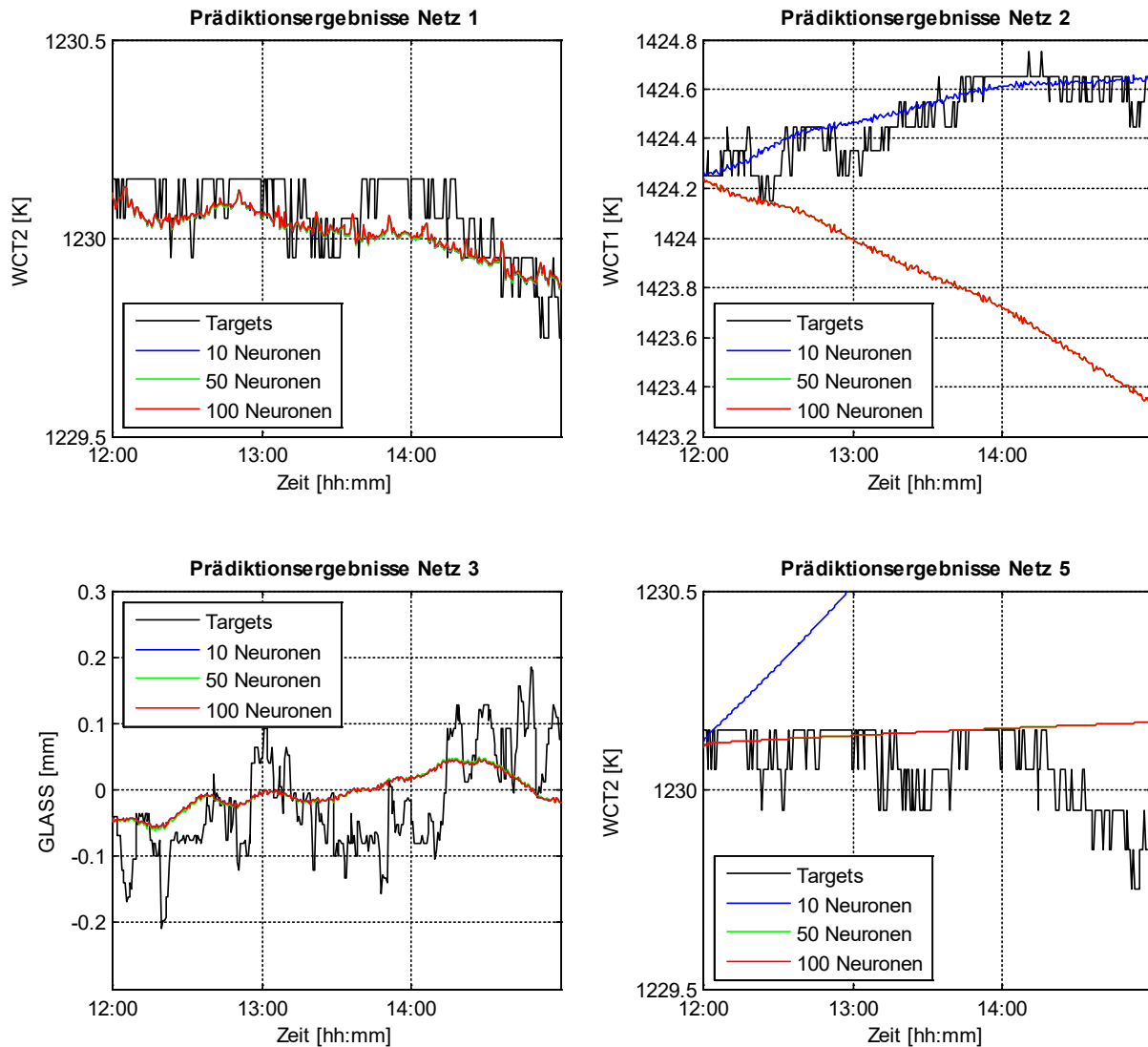


Abbildung 4.2: Prädiktionsergebnisse zum Einfluss der Neuronenanzahl

#### 4.2.2 Einfluss der Tapped Delay Lines

Die Tapped Delay Lines besitzen einen erheblichen Einfluss auf die Prädiktionsergebnisse. In Abbildung 4.3 sind die berechneten Zeitverläufe von Netz 1 und 3 für verschiedene TDL-Werte dargestellt. Bei Netz 1 (links) wird deutlich, dass die Charakteristik des tatsächlichen Ausgangssignals (schwarz dargestellt) mit höheren TDL-Werten besser angenommen werden kann. Die Standardabweichung für TDL = 60 beträgt im dargestellten Prädiktionszeitraum 0,0729 K und liegt damit nur 0,0108 K über der Standardabweichung der OSP bei gleicher Konfiguration. Zu hohe TDL-Werte (z. B. TDL = 120) führen zu einer sichtbar verschlechterten Qualität der Ausgangsgröße. Bei Netz 3 führt dieser hohe TDL-Wert sogar zu einem Ausreißer.

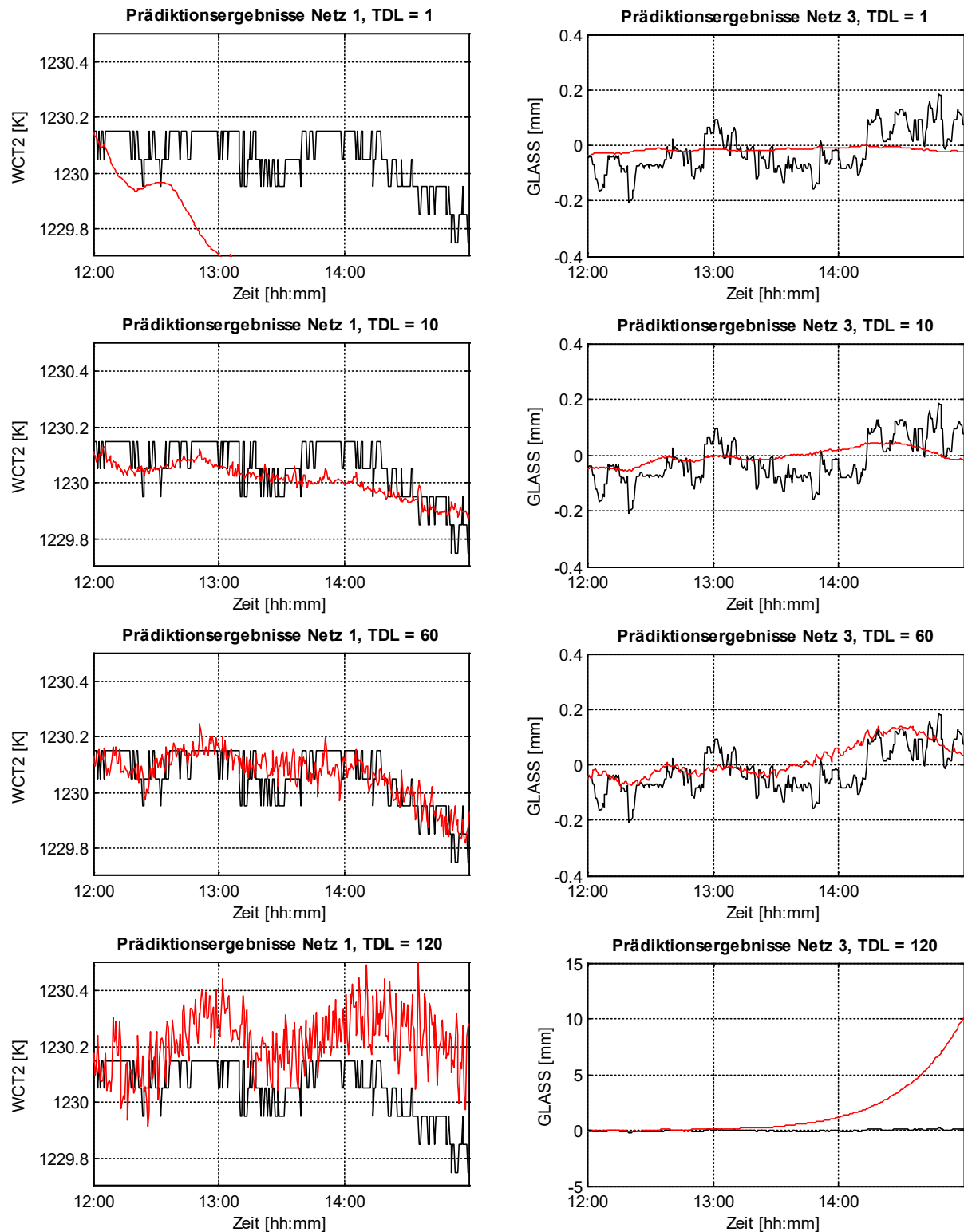


Abbildung 4.3: Prädiktionsergebnisse von Netz 1 und 3 zum Einfluss der TDL  
(Bem.: Netz und TDL-Wert sind Titel des jeweiligen Diagramms)

Auch bei dieser Untersuchung sind die Simulationsergebnisse, mit Ausnahme von Netz 1, in Summe nicht zufriedenstellend. Der Verlauf des Glasstands (Netz 3) kann mit keiner TDL-Konfiguration ausreichend genau berechnet werden. Gleiches gilt für die Netze 2 und 5.

Einen direkten Vergleich der Prädiktionsergebnisse der OSP und MSP liefert Abbildung 4.4 am Beispiel von Netz 1. Der optimale TDL-Wert ist in der Legende angegeben. Um eine annähernd so hohe Genauigkeit wie bei der OSP zu erzielen, ist bei Netz 1 bereits ein TDL-Wert von 40 notwendig. Die Standardabweichung im Prädiktionszeitraum ist im Vergleich zur OSP dennoch um 37 % höher (Tabelle 4.1).

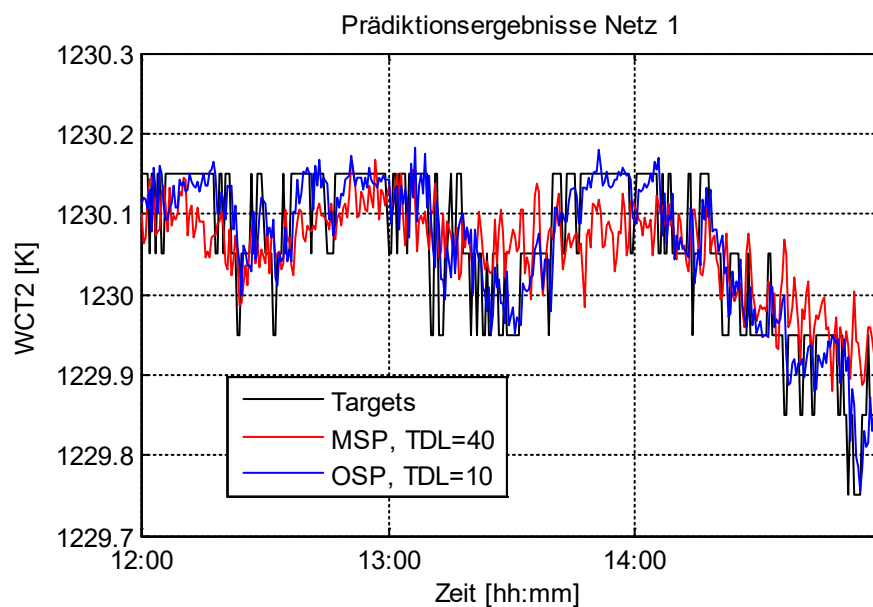


Abbildung 4.4: Vergleich der Prädiktionsergebnisse für OSP und MSP bei optimaler Konfiguration von Netz 1

Tabelle 4.1: Vergleich zwischen OSP und MSP für Netz 1 im dargestellten Prädiktionszeitraum (Abbildung 4.4)

Netz	Kennzahl	OSP	MSP	Differenz
1	max. Spannweite aller Fehler	0,3069 K	0,4202 K	+37 %
	Betrag des arithm. Mittelwerts aller Fehler	0,0003 K	0,0023 K	+667 %
	Standardabweichung aller Fehler	0,0514 K	0,0719 K	+ 40 %



### 4.2.3 Einfluss des Trainings- und Prädiktionszeitraums

Zu dieser Untersuchung wurden die Trainings- und Prädiktionszeiträume variiert. Aufgrund der bisherigen Erkenntnisse zur MSP beschränkt sich die Auswertung allein auf Netz 1.

Zuerst wird der Einfluss des Trainingszeitraums erläutert. In der Abbildung 4.5 sind für zwei Prädiktionszeiträume die berechneten Verläufe der Ausgangsgröße WCT2 für verschiedene Längen des Trainingszeitraums dargestellt. Ein Trainingszeitraum von zwölf Stunden führt im ersten Prädiktionszeitraum zu einem akzeptablen Ergebnis. Der maximale Fehler beträgt lediglich 0,2536 K, die Standardabweichung 0,0719 K. Eine Verkürzung der Trainingszeiträume führt zu einer sichtbaren Reduzierung der Genauigkeit. Der arithmetische Fehler-Mittelwert steigt von 0,0023 K (12 h Training) auf 0,1881 K (6 h Training). Im zweiten Prädiktionszeitraum entsprechen die Ergebnisse nicht den Erwartungen. Hier wird mit einem sechsständigen Trainingszeitraum die höchste Genauigkeit erzielt. Das rechte Diagramm ist repräsentativ für die restlichen, nicht dargestellten Simulationsergebnisse. Ein Vergleich aller zeitlicher Verläufe lässt keine allgemeingültige Aussage zur Abhängigkeit des Trainingszeitraums in Bezug auf die MSP zu.

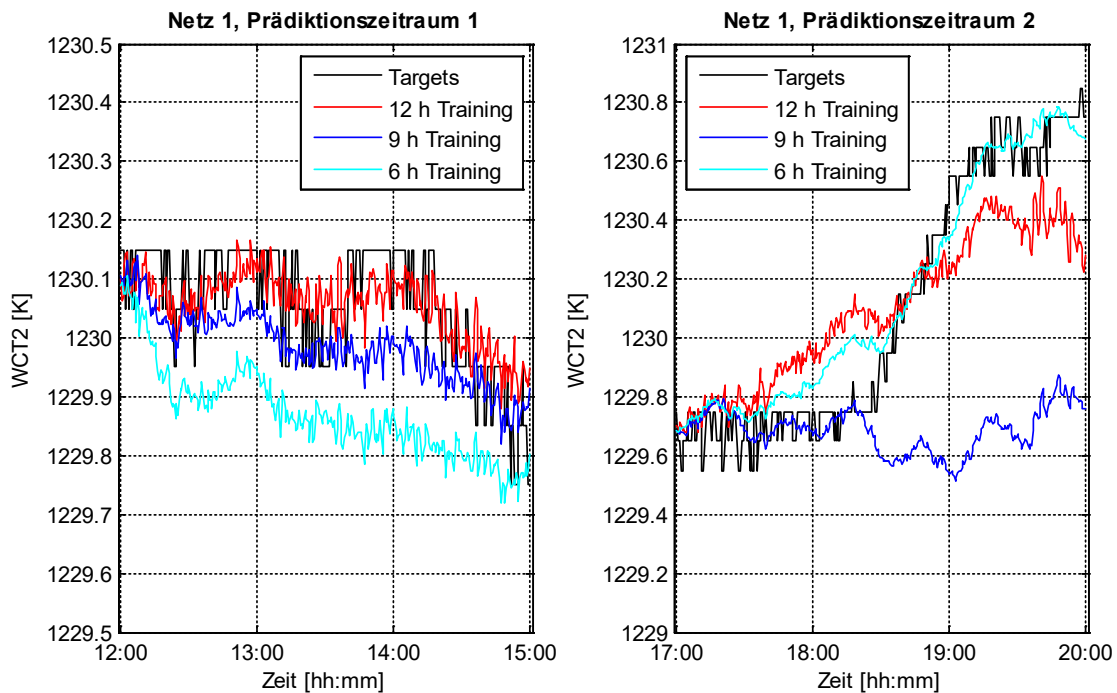


Abbildung 4.5: Prädiktionsergebnisse von Netz 1 in Abhängigkeit der Länge des Trainingszeitraums

Nachfolgend wird der Einfluss des Prädiktionshorizonts beschrieben. Um eine Abhängigkeit zum tatsächlichen Verlauf der Ausgangsgröße zu vermeiden, wurde der Prädiktionszeitraum von 12:00 - 21:00 Uhr schrittweise um eine Stunde verschoben. Eine Mittelwertbildung aller berechneten Standardabweichungen zeigt eine lineare Abhängigkeit (Abbildung 4.6). Je länger der Prädiktionszeitraum, desto mehr weicht der berechnete vom tatsächlichen Verlauf ab. Diese Tendenz ist unabhängig von der Länge des Trainingszeitraums.

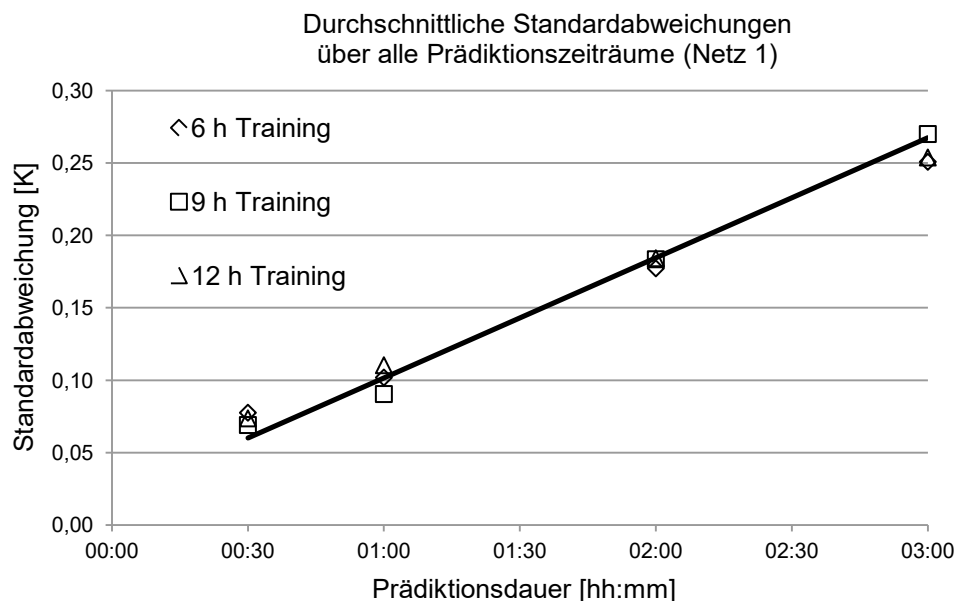


Abbildung 4.6: Abhängigkeit der Standardabweichungen bei verschiedenen Trainings- und Prädiktionshorizonten

Die Standardabweichung in den ersten 30 Minuten im Prädiktionszeitraum beträgt bei zwölfstündigem Trainingszeitraum im Durchschnitt 0,0737 K. Dieser Wert wurde aus allen Prädiktionszeiträumen gebildet und liefert somit einen Richtwert, mit welcher Standardabweichung durchschnittlich in den ersten 30 Minuten gerechnet werden kann.

Eine Möglichkeit um geänderte dynamische Zusammenhänge im KNN zu berücksichtigen ist ein in regelmäßigen Abständen erneutes Training. In Abbildung 4.7 wird ein mögliches Szenario dargestellt, wie sich die Prädiktionsergebnisse unterscheiden können, wenn das Netz zu jeder vollen Stunde neu trainiert und eine einstündige Prädiktion angesetzt wird. Der Trainingszeitraum beträgt bei diesem Beispiel jeweils zwölf Stunden und endet unmittelbar vor Prädiktionsbeginn.

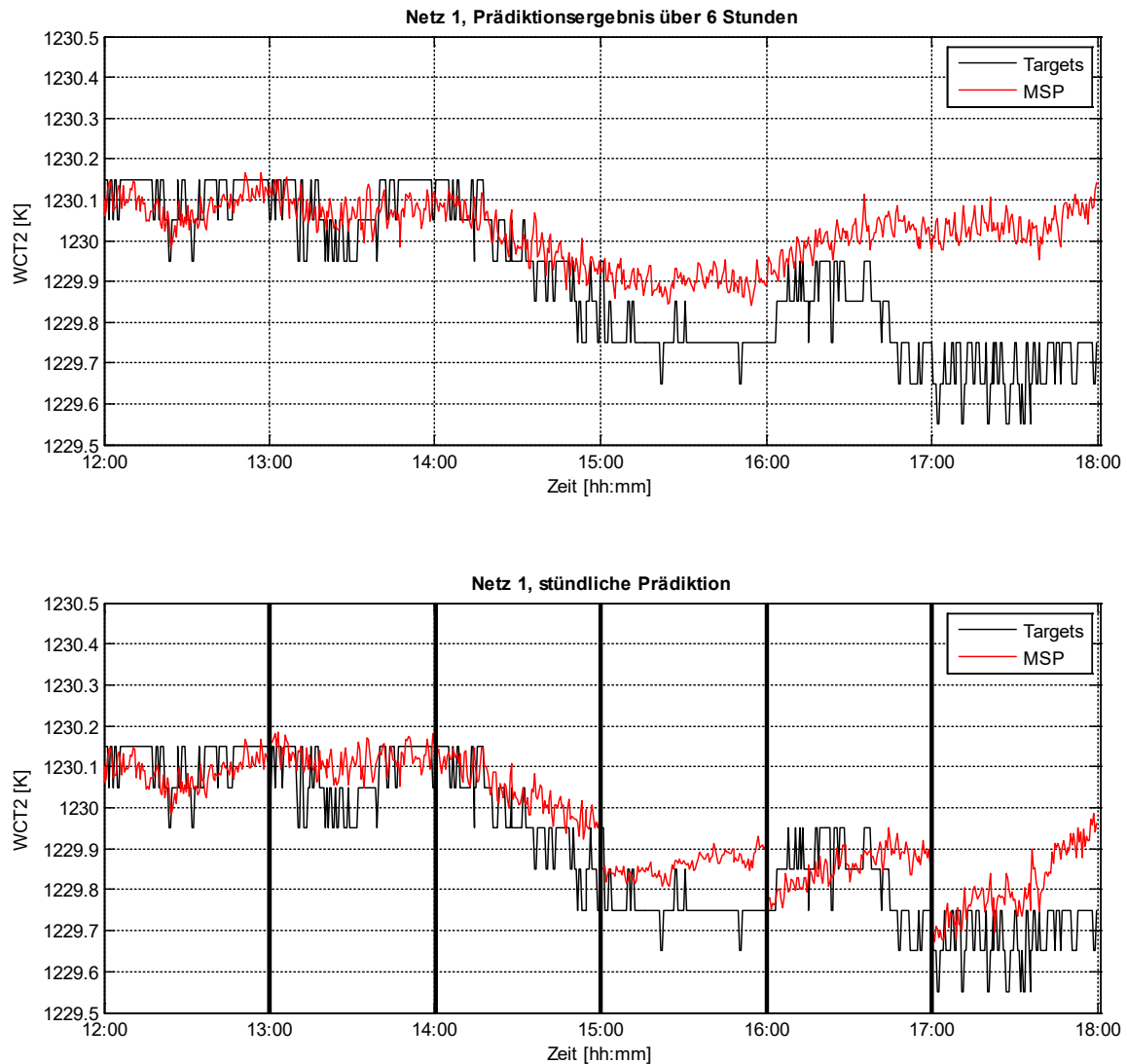


Abbildung 4.7: Vergleich einer sechsständigen mit einer stündlich neu durchgeführten Prädiktion am Beispiel von Netz 1

Die "gleitende" Simulation führt zu einem etwas besseren Ergebnis. Bis 15:00 Uhr unterscheiden sich die Zeitverläufe beider Varianten nur geringfügig. Ab 15:00 Uhr ist zu erkennen, dass das stündlich neu trainierte Netz die Prädiktion in der Nähe vom tatsächlichen Verlauf beginnt. Allerdings weichen die Verläufe im Anschluss vergleichsweise schnell ab.

### 4.3 Zusammenfassung

Die in Kapitel 3 evaluierten KNN wurden durch eine gezielte Änderung der Netzstruktur zur MSP konvertiert. Es erfolgte eine stichprobenartige Untersuchung zum Einfluss ausgewählter Netzparameter. Die Prädiktionsergebnisse sind im Vergleich zur OSP signifikant schlechter. Lediglich Netz 1 erzielt unter bestimmten Voraussetzungen akzeptable Resultate

und kommt für eine prädiktive Regelstrategie infrage. Nachfolgend sind die Kernaussagen zur Multi Step Prediction zusammengefasst.

- Die Anzahl der Neuronen hat keinen Einfluss auf die berechneten Ausgangsgrößen, womit die gewonnenen Erkenntnisse zur OSP in diesem Fall bestätigt werden (vgl. Abschnitt 3.2.4). Mitunter treten vereinzelte Ausreißer auf, die auf numerische Ungenauigkeiten zurückgeführt werden können.
- Die Anzahl der Tapped Delay Lines beeinflusst maßgeblich die Charakteristik der Ausgangsgröße. Die Schwankungsbreite der Amplituden nimmt mit steigenden TDL-Werten zu. Der Verlauf kann zunehmend an die Zielgröße angenähert werden. Zu hohe TDL-Werten führen vereinzelt zu Ausreißern.
- Die Länge des Trainingszeitraums hat keinen allgemeingültigen Einfluss auf die anschließenden Prädiktionsergebnisse.
- Mit zunehmender Prädiktionsdauer steigen die Standardabweichungen proportional an (vgl. Abbildung 4.6).
- Scheinbar passende Konfigurationen können für andere Prädiktionszeiträume inakzeptable Ergebnisse liefern. Lediglich Netz 1 ist für die MSP geeignet. Mit optimaler Konfiguration (10 Neuronen, TDL = 10, 12 h Training) beträgt die Standardabweichung in den ersten 30 Minuten der Prädiktion durchschnittlich 0,0737 K.

Die Simulationsergebnisse hängen maßgeblich vom Verlauf und vom Informationsgehalt der verwendeten Daten im Trainings- und Prädiktionszeitraum ab. Insbesondere der Auswahl des Trainingsdatensatzes kommt eine besondere Bedeutung zu. Die Ergebnisse verbessern sich für alle fünf Netze erheblich, wenn die Prädiktion im Trainingszeitraum durchgeführt wird. Diese Simulation, deren Ergebnisse nicht dargestellt sind, ist praxisuntauglich, untermauert aber die Einschränkung der KNN nur gelerntes Prozesswissen hinreichend genau abbilden zu können.

### **Bemerkungen**

Die mangelhaften Ergebnisse der MSP bedeuten nicht zwangsläufig eine geminderte Eignung für prädiktive Regelstrategien. Zwischen der in diesem Kapitel beschriebenen MSP und einer prädiktiven Regelung bestehen folgende Unterschiede.

- Bei der MSP nutzt das Netz über den gesamten Prädiktionszeitraum die tatsächlichen Eingangsgrößen. Zusätzlich werden die eigenen berechneten Ausgangsgrößen zurückgeführt, womit das Thema der Fehlerfortpflanzung eine besondere Rolle spielt. Mit steigender Prädiktionsdauer nehmen die Fehler zu. Die Ursache liegt

darin, dass die Informationen zu den Ein- und Ausgangsgrößen nicht mehr zu den Informationen passen, mit denen das Netz trainiert wurde. Das Netz kennt die Eingangsgrößen und könnte mit den tatsächlich zurückliegenden Ausgangsgrößen die zukünftige Ausgangsgröße sehr genau berechnen. Die eigenen berechneten und zurückgeführten Ausgangsgrößen weichen aber bei der MSP zunehmend ab und passen nicht mehr zu den angelegten Eingangsgrößen. Das Netz kann diese Werte nicht immer verarbeiten. Das ist die Ursache, warum mitunter Ausreißer entstehen (vgl. Abbildung 4.3, Netz 3, TDL = 120).

- Bei der prädiktiven Regelung verwendet das Netz zur Prädiktion stets die tatsächlich zurückliegenden Messwerte. Die zukünftigen Eingangsgrößen stehen nicht zur Verfügung, sie sind Lösungsbestandteil der Optimierungsaufgabe. Zudem erfolgt die Prädiktion rekursiv, d. h. die Netze können in der "open loop"-Struktur bleiben. Der gesamte Optimierungsalgorithmus wird zum nächsten Abtastzeitpunkt erneut durchgeführt und das Netz erhält die tatsächlichen Messdaten.

Mit einer vergleichsweise kurzen Prädiktionsdauer (z. B.  $< 1$  h) sind aus den genannten Gründen die evaluierten Netze, insbesondere das NARX-Netz 1 und das NIO-Netz 4, für prädiktive Regelaufgaben geeignet.

---

## 5 Regelungstechnisches Konzept

Mit den evaluierten Künstlichen Neuronalen Netzen stehen Prozessmodelle zur Verfügung, die für den Entwurf einer Regelung infrage kommen. In diesem Kapitel werden mögliche Regelstrukturen mit den Anforderungen abgeglichen, denen das Regelkonzept gerecht werden muss. Im zweiten Abschnitt wird das Konzept der neuronalen prädiktiven Regelung beschrieben. Neben der mathematischen Beschreibung zur Prädiktion der Regelgrößen wird eine Auswahl eines geeigneten Gütekriteriums und Optimierungsverfahrens getroffen. Abschließend werden Grenzen und mögliche Alternativen diskutiert.

### 5.1 Ableitung einer geeigneten Regelstrategie

Die Verwendungsmöglichkeiten von KNN für Regelungszwecke sind sehr vielfältig. Eine auf Basis von [64, 67, 70 - 72] erstellte Übersicht zeigt die Verknüpfungen Künstlicher Neuronaler Netze hinsichtlich bestehender Regelkonzepte (Abbildung 5.1). Die Übersicht erhebt keinen Anspruch auf Vollständigkeit, vielmehr gibt sie einen geordneten Überblick zu infrage kommenden Regelkonzepten.

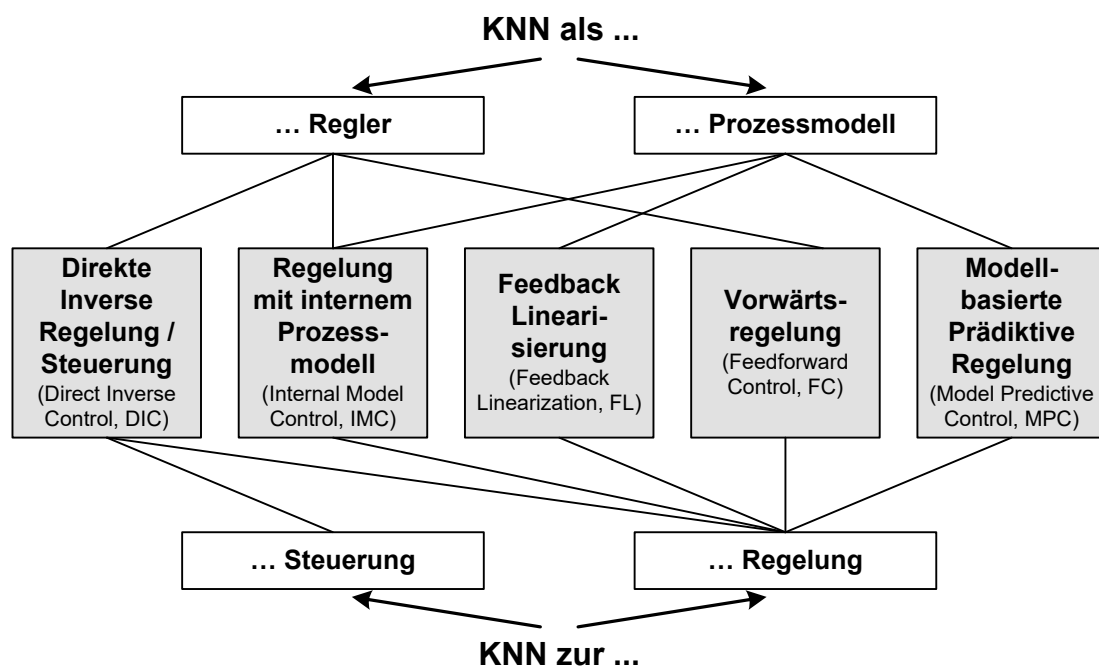


Abbildung 5.1: Verwendungsmöglichkeiten von KNN für Regelungszwecke

Die Auswahl der Regelstrategie hängt im hohen Maße von den Prozesseigenschaften und den daraus resultierenden Anforderungen ab. Nachfolgend sind die für den gegebenen Anwendungsfall abgeleiteten Anforderungen aufgelistet.

- **Akzeptabler Berechnungsaufwand**

Die Abtastrate des verwendeten Datensatzes beträgt 30 Sekunden. In dieser Zeit müssen die Berechnungen mit den KNN abgeschlossen sein.

- **Prädiktionsfähigkeit**

Die signifikant großen Zeitverzögerungen im Schmelzofen erschweren das Verständnis, wie sich aktuelle Stell- und nicht messbare Störgrößen auf zukünftige Werte der Regelgrößen auswirken. Aus diesem Grund sollte das Regelgesetz prognostizierte Regelgrößen und -abweichungen nutzen.

- **Mehrgrößentauglichkeit**

Die evaluierten Netze besitzen stets nur eine Ausgangsgröße, aber zum Teil mehrere Eingangsgrößen. Das Regelkonzept muss dementsprechend für den Mehrgrößenfall geeignet sein.

- **Stabilität**

Der geschlossene Regelkreis, sei es mit KNN als Regler oder als Prozessmodell, muss stabil sein. Die Stabilität von KNN kann nicht durch konventionelle Stabilitätskriterien überprüft werden. Im übertragenen Sinne sind KNN stabil, wenn sie keine oszillierenden oder verrauschten Ausgangswerte liefern.

- **Ausreichende Anzahl an Reglerparametern**

Neuronale Regler besitzen nicht die üblichen Reglerparameter wie PID-Regler. Sie können zwar Nichtlinearitäten und Störungen kompensieren, ein gewünschtes Regelkreisverhalten kann an einem KNN aber nicht explizit eingestellt werden. Das Regelkonzept muss zusätzliche Parameter bieten, mit denen Einfluss auf das Regelkreisverhalten genommen werden kann.

- **Adaptivität**

Unter diesem Begriff ist nicht die Fähigkeit gemeint, dass sich die KNN an ein sich änderndes Prozessverhalten anpassen, sondern vielmehr die Möglichkeit des Regelkonzeptes, im laufenden Prozess Änderungen zu verwirklichen. Dazu zählt das Ändern oder Hinzufügen von Nebenbedingungen (z. B. Stellgrößenbeschränkungen) oder das Verwenden eines neu trainierten Netzes, ohne den Regler neu konfigurieren zu müssen.

- **Störungsresistenz**

Beim Glasherstellungsprozess treten mehrere nicht messbare Störungen auf, u. a. der Verschleiß des Feuerfestmaterials, Schwankungen in der Gemenge-Zusammensetzung oder Falschlufteinbrüche. Das Regelkonzept sollte nicht messbare Störungen und Modellunsicherheiten berücksichtigen.

Nachfolgend werden die in der Abbildung 5.1 dargestellten Regelstrukturen kurz beschrieben und hinsichtlich der aufgezählten Anforderungen bewertet.

### **Direct Inverse Control (DIC)**

Bei der Direkten Inversen Regelung wird ein inverses Prozessmodell trainiert, das in eine Steuerung oder Regelung implementiert werden kann. Dieses Prozessmodell kann direkt als neuronaler Regler oder in Kombination mit einem konventionellen Regler genutzt werden. Auf diese Weise werden Nichtlinearitäten der Strecke kompensiert. Das Training der Netze erfolgt entweder offline (Ziel: Minimierung Modellfehler) oder online (Ziel: Minimierung Regelabweichung). DIC stellt eine sehr einfache Regelstruktur dar und ist intuitiv verständlich. Der Berechnungsaufwand ist im Vergleich zu anderen Regelstrukturen gering. Die echtzeitfähige Methode ist einfach zu implementieren und für zeitvariante Systeme geeignet, denn ein erneutes Training des KNN ist jederzeit möglich. Probleme treten auf, wenn MIMO- oder MISO-Netze verwendet werden und die inversen Prozessmodelle zur Instabilität neigen. Wird kein konventioneller Regler verwendet, gibt es auch keine Einstellparameter, mit denen Einfluss auf das geschlossene Regelkreisverhalten genommen werden kann.

### **Internal Model Control (IMC)**

Bei dieser Struktur sind zwei KNN im Einsatz. Ein Netz dient als Prozessmodell, ein anderes als Regler. Der Unterschied zu einem konventionellen Regelkreis besteht in der Rückführung. Es wird nicht der Istwert der Regelgröße, sondern die Abweichung zwischen der tatsächlichen und der vom KNN berechneten Regelgröße zurückgeführt, die somit einen Schätzwert der nicht messbaren Störungen und der Modellunsicherheit darstellt. Der neuronale Regler bildet das inverse Streckenverhalten ab und ist dadurch in der Lage Störungen und Nichtlinearitäten zu kompensieren. Die Regelung ist stabil, wenn beide Netze und der Prozess stabil sind, was zugleich eine große Einschränkung bedeutet. Als Reglerparameter kann ein zusätzliches Filter eingesetzt werden. Der Rechenaufwand bei der IMC-Struktur ist minimal höher, da mit zwei KNN gerechnet werden muss. Die Mehrgrößentauglichkeit und Prädiktionsfähigkeit ist nur mit hohem Aufwand zu realisieren. Zudem ist es schwierig, Nebenbedingungen zu berücksichtigen.



**Feedback Linearization (FL)**

Allgemeines Ziel dieser Methode ist es, die Nichtlinearitäten der Regelstrecke durch Linearisierung zu eliminieren. Ein oft verwendeter Ausgangspunkt ist die Beschreibung der Regelstrecke in Zustandsraumdarstellung (Regelungsnormalform). Mit KNN ändert sich die Regelstruktur nur geringfügig. Es werden zwei KNN benötigt, von denen ein Netz auf Basis zurückliegender Ein- und Ausgangsgrößen das dynamische Verhalten abbildet. Dieses Netz repräsentiert die nichtlineare Funktion der Regelstrecke. Das andere Netz stellt das inverse Streckenverhalten dar, wird aber nicht direkt als Regler verwendet. Das Regelgesetz ergibt sich durch Umstellen der Gleichungen nach der Stellgröße, vergleichbar mit der klassischen Zustandsrückführung. Durch eine geeignete Polvorgabe kann das Regelkreisverhalten entsprechend angepasst werden. Weiterhin ist keine Anpassung des Regelgesetzes notwendig, wenn die KNN neu trainiert werden. Da aber auch hier ein inverses Streckenmodell benötigt wird, kann die Stabilität, insbesondere nach neuem Training, beeinträchtigt werden. Eine weitere Schwierigkeit besteht in der Wahl der Modellstrukturen.

**Feedforward Control (FC)**

Das KNN wird einem konventionellen Regler parallel geschaltet und berechnet anhand der vorgegebenen Sollwerte und zurückliegenden Stellgrößen ein Stellsignal, das dem eigentlichen Stellsignal des Reglers aufsummiert wird. Diese Struktur ist besonders für eine Folgeregelung geeignet. Ist die Sollwerttrajektorie bekannt, können die Stellgrößen schon vorab berechnet werden. Die Stabilität und die Störunterdrückung wird durch den PID-Regelkreis gesichert. Ein weiterer Vorteil ist, dass der KNN-Regler adaptiv zugeschaltet werden kann, insbesondere bei Sollwertänderungen. Eine Voraussetzung zur Verwendung dieser Struktur ist eine bestehende Feedback-Regelung. Die Anzahl der Reglerparameter ist demzufolge ausreichend. Eine Schwierigkeit besteht in dem Training des Netzes. Ist der zukünftige Sollwertverlauf über einen längeren Zeitraum konstant, so dürften die vom Netz berechneten Ausgangsgrößen wenig aussagekräftig sein. Bei einem schlecht konditionierten Netz besteht die Gefahr, dass sich das Regelkreisverhalten verschlechtert. Die Mehrgrößentauglichkeit ist stark eingeschränkt, da beide Stellgrößen aufsummiert werden.

**Model Predictive Control (MPC)**

Dieser Begriff stellt eine Sammlung von mehreren Regelstrukturen dar, die allesamt auf dem Prinzip basieren, in einem zukünftigen Prädiktionshorizont ein Gütekriterium zu minimieren. Zur Minimierung des Gütekriteriums sind Optimierungsverfahren notwendig, deren Wahl vom jeweiligen Anwendungsfall abhängt. Das neuronale Prozessmodell wird ausschließlich zur Prädiktion zukünftiger Regelgrößen genutzt. Dieses Regelprinzip kann besonders bei Systemen mit sehr großen Zeitverzögerungen angewandt werden. Der kriteriumbasierte

Ansatz ist intuitiv verständlich und erlaubt die Implementierung von Nebenbedingungen. Damit können zusätzliche Stabilitäts- und Sicherheitsanforderungen berücksichtigt werden. Der Optimierungsalgorithmus bietet mehrere Parameter, mit denen Einfluss auf das Regelkreisverhalten genommen werden kann.

Einen zusammenfassenden Abgleich der beschriebenen Regelstrukturen bezüglich der Prozesskompatibilität liefert folgende Tabelle.

Tabelle 5.1: Abgleich bestehender Regelstrukturen mit gestellten Anforderungen zur Bewertung der Prozesskompatibilität

	Regelstruktur				
	Direct Inverse Control (DIC)	Internal Model Control (IMC)	Feedback Linearization (FL)	Feedforward Control (FC)	Model Predictive Control (MPC)
<u>Bedeutung der Symbole:</u> ++ = Anforderung wird sehr gut erfüllt + = Anforderung wird gut erfüllt o = vom Anwendungsfall abhängig - = Anforderung wird kaum erfüllt -- = Anforderung wird nicht erfüllt					
Anforderungen					
Akzeptabler Berechnungsaufwand	++	+	o	+	-
Prädiktionsfähigkeit	-	-	-	-	++
Mehrgrößentauglichkeit	-	-	-	--	o
Stabilität	o	+	o	++	+
Ausreichende Anzahl an Reglerparametern	-	-	-	++	+
Adaptivität	-	-	+	o	++
Störungsresistenz	o	+	o	o	o

Für den gegebenen Anwendungsfall ist die Modellbasierte Prädiktive Regelung (MPC) am besten geeignet. Mit dem Prinzip der "Regelung durch Optimierung" lassen sich Nebenbedingungen und Stabilitätsanforderungen berücksichtigen. Das neuronale Prozessmodell wird ausschließlich zur Prädiktion verwendet. Die Schwierigkeiten beim Umgang mit inversen Streckenmodellen sind somit nicht existent. Aufgrund der großen Zeitverzögerungen bietet sich die prädiktive Regelung besonders an. Die Optimierungsaufgabe wird zu jedem Abtastzeitschritt gelöst, womit aktuelle Istwerte der Regelgrößen (und damit auch nicht messbare Störungen) einbezogen werden. Um auf eventuelle Änderungen der Prozessdynamik und

der Umgebungsbedingungen zu reagieren, können die KNN während der Betriebsphase neu trainiert werden, ohne dass Anpassungen am Regelgesetz notwendig sind. Der wesentliche Nachteil ist der Berechnungsaufwand. Die Optimierungsaufgabe muss innerhalb von 30 Sekunden gelöst sein. Damit kommt den Reglerparametern (u. a. der Auswahl des Prädiktionshorizontes) eine besondere Bedeutung zu.

Bereits heute werden modellbasierte prädiktive Ansätze in der Glasindustrie verwendet, z. B. in [40, 41, 44, 73]. In der Regel dienen Sprungantwortmodelle oder PT1-Übertragungsfunktionen als Prozessmodelle. Bestehende MPC-Regelungen auf Basis von KNN sind in der Glasindustrie nach aktuellem Kenntnisstand nicht bekannt. In anderen Industriezweigen findet diese Kombination bereits Anwendung. So werden z. B.

- die Temperaturen in einem Härtingsofen für Kaltwalzen [72] oder
- die Einschweißtiefe beim Laserstrahlschweißen [54]

mit KNN prädiktiv geregelt. Weitere Beispiele finden sich u. a. in [67, 74, 75].

## 5.2 Neuronale prädiktive Regelung

### 5.2.1 Grundprinzip

Die prinzipielle Wirkungsweise modellbasierter prädiktiver Regelungen wird ausführlich in [73, 76] beschrieben. In [73] findet sich ein Beispiel für eine MPC-Regelung in einem Feeder-Kanal (Glasrinne bei Behälterglaswannen). Das Grundprinzip einer MPC-Regelung mit neuronalen Prozessmodellen wird sehr ausführlich in [67] und [70] beschrieben. Der Komplexität des aufzustellenden Regelkonzeptes sind nach oben keine Grenzen gesetzt. Der MPC-Ansatz kann sowohl für einfache Eingrößenregelungen als auch für eine Mehrgrößenregelung angewandt werden. Eine weitere Möglichkeit besteht darin, ein übergeordnetes Gütekriterium zu wählen (z. B. den Energieverbrauch der gesamten Schmelzanlage). Der MPC-Regler hat dann die Aufgabe, für die unterlagerten Standard-PID-Basisregelungen die Sollwerte vorzugeben, mit denen das Gütekriterium erfüllt werden kann. Dieser Ansatz wird bereits beim *Expert System ESIII* der Firma GlassService Inc. (Tschechien) verwendet [41].

Im Rahmen dieser Arbeit wird das Konzept einer einfachen MPC-Regelung aufgestellt, bei der eins der evaluierten KNN verwendet und die jeweilige Regelgröße prädiktiv geregelt werden kann. Das vereinfachte Grundprinzip ist in Abbildung 5.2 dargestellt.

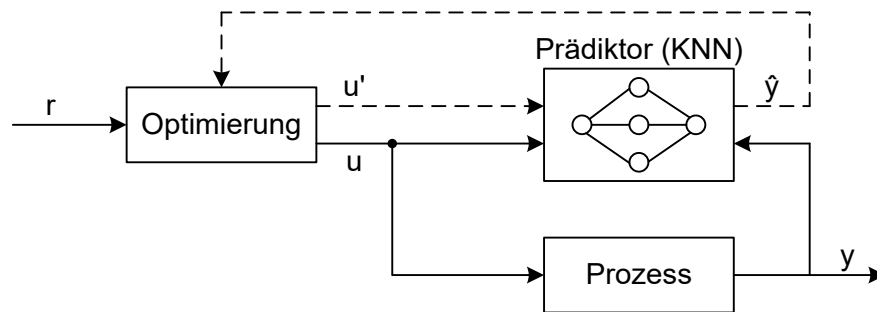


Abbildung 5.2: Grundprinzip der neuronalen prädiktiven Regelung (Basis: [67])

Das KNN berechnet auf Basis zukünftiger zu optimierender Stellgrößen  $u'$  die Vorhersage der Regelgrößen  $\hat{y}$  für einen definierten Prädiktionshorizont. Ziel der Optimierung ist es, das vorgegebene Gütekriterium zu minimieren und daraus die optimalen zukünftigen Stellgrößen  $u'$  zu berechnen. Das Ergebnis ist eine Folge von Stellgrößen, von der lediglich die erste Stellgröße  $u$  an den Prozess ausgegeben wird. Die betrachteten Zeithorizonte verschieben sich fortwährend und die Optimierungsaufgabe wird zu jedem Abtastschritt neu gelöst. Diese Herangehensweise wird in der Fachliteratur Prinzip des "gleitenden Horizonts" genannt und stellt sicher, dass stets aktuelle Informationen der Regel- und Stellgrößen einbezogen werden. Ohne diesen Ansatz beruht die berechnete Stellgrößenfolge auf veralteten Informationen. Eine Reaktion auf nicht messbare Störungen wäre erst nach dem Ablauf des berechneten Prädiktionshorizontes (bzw. Stellhorizontes) möglich.

Zur Realisierung dieser neuronalen prädiktiven Regelung sind

- ein geeignetes KNN als Prädiktor,
- ein Gütekriterium und
- ein geeigneter Optimierungsalgorithmus

notwendig, die in den folgenden Abschnitten beschrieben werden.

## 5.2.2 Mathematische Beschreibung des Prädiktors

### 5.2.2.1 One Step Prediction

Die Gleichungen zur Berechnung des zukünftigen Ausgangswertes hängen von der Netzstruktur und -konfiguration ab. Die KNN besitzen zwei Schichten. Die Ausgangswerte der einzelnen Neuronen der ersten Schicht sind gleichzeitig die Eingangswerte der zweiten Schicht. Nach Abbildung 5.3 berechnet sich der Ausgangswert  $a$  eines Neurons wie folgt:

$$n = w_1 u_1 + w_2 u_2 + \dots + w_R u_R + b = \left( \sum_{i=1}^R w_i u_i \right) + b \quad (5.1)$$

$$a = f_{akt}(n) \quad (5.2)$$

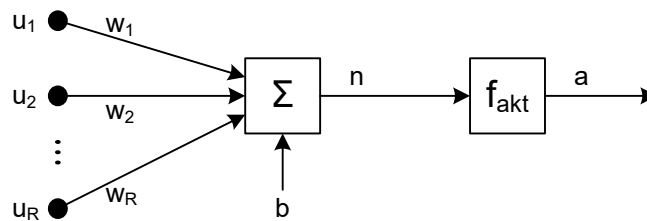


Abbildung 5.3: Blockschaltbild für ein Neuron mit  $R$  Eingangsgrößen

Der Eingangswert  $n$  des Neurons setzt sich aus den mit  $w_i$  gewichteten Eingangsgrößen  $u_i$  und dem Schwellwert  $b$  zusammen. Im Vergleich zur Aktivierungsfunktion *tansig* vereinfacht sich die Gleichung für den Ausgangswert bei linearer Aktivierungsfunktion sichtlich.

$$\text{für } \textit{tansig}: \quad a = \tanh(n) = \frac{1 - e^{-2n}}{1 + e^{-2n}} \quad (5.3)$$

$$\text{für } \textit{purelin}: \quad a = \textit{purelin}(n) = n \quad (5.4)$$

Für die Gleichung (5.1) folgt in Matrizenschreibweise

$$\begin{pmatrix} n_1 \\ \vdots \\ n_{S^1} \end{pmatrix} = \begin{pmatrix} w_{1,1} & \cdots & w_{1,R} \\ \vdots & \ddots & \vdots \\ w_{S^1,1} & \cdots & w_{S^1,R} \end{pmatrix} \cdot \begin{pmatrix} u_1 \\ \vdots \\ u_R \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_{S^1} \end{pmatrix} \quad \text{bzw.} \quad \vec{n} = W \cdot \vec{u} + \vec{b}. \quad (5.5)$$

In der verborgenen Schicht sind die Wichtungsfaktoren, die mit den Eingangsgrößen  $u$  multipliziert werden, mit  $iw$  gekennzeichnet. Die Wichtungsfaktoren in der Ausgangsschicht lauten  $lw$ . Zur eindeutigen Zuordnung der Wichtungsfaktoren werden Indizes eingeführt. Der erste tiefgestellte Index steht für das Neuron, der zweite für die Eingangsgröße. Der erste hochgestellte Index gibt die Schicht der Wichtungsfaktoren an, der zweite die Schicht, aus welcher die Signale stammen. Abbildung 5.4 stellt das Blockschaltbild eines zweischichtigen Netzes mit  $R$  Eingangsgrößen,  $S^1$  verborgenen Neuronen und einer Ausgangsgröße  $y$  dar.

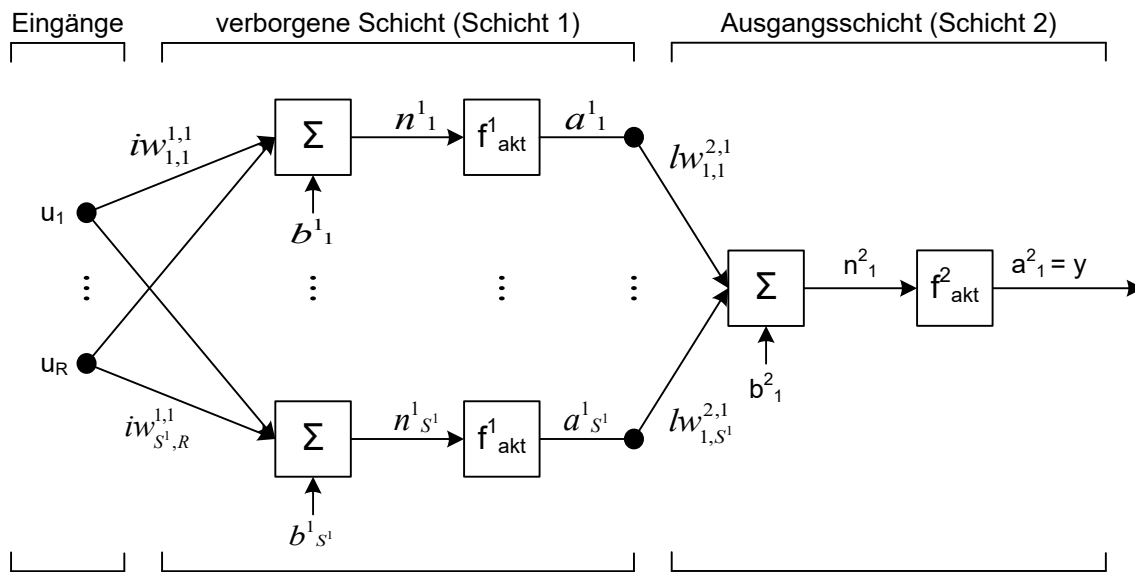


Abbildung 5.4: Blockschaltbild eines zweischichtigen Netzes mit  $R$  Eingängen,  $S^1$  Neuronen und einem Ausgang

Mit der Annahme, dass ausschließlich lineare Aktivierungsfunktionen verwendet werden, berechnet sich der Eingangswert  $n_1^2$  des Neurons in der Ausgangsschicht nach

$$n_1^2 = a_1^1 \cdot lw_{1,1}^{2,1} + \dots + a_{S^1}^1 \cdot lw_{1,S^1}^{2,1} + b_1^2 = \left( \sum_{j=1}^{S^1} a_j^1 \cdot lw_{1,j}^{2,1} \right) + b_1^2. \quad (5.6)$$

Die Ausgangsgröße  $y$  ergibt sich durch Anwenden der Gleichungen (5.1), (5.4) und (5.6) zu

$$y = a_1^2 = \left[ \sum_{j=1}^{S^1} lw_{1,j}^{2,1} \cdot \left( \sum_{i=1}^R iw_{j,i}^{1,1} \cdot u_i \right) + b_j^1 \right] + b_1^2, \quad (5.7)$$

$$y = LW^{2,1} \cdot (IW^{1,1} \cdot \vec{u} + \vec{b}^1) + b^2. \quad (5.8)$$

Für NARX-Netze fehlen noch die Terme der zurückgeführten Ausgangsgrößen und der Tapped Delay Lines. Angelehnt an Abbildung 5.4 zeigt folgende Abbildung das um die TDL-Werte erweiterte Blockschaltbild. Die Notation der nun zeitabhängigen Größen wird aus Gründen der Übersichtlichkeit wie folgt festgelegt:  $u_1(t-d) = u_{1,d}$  mit  $1 \leq d \leq TDL$ .

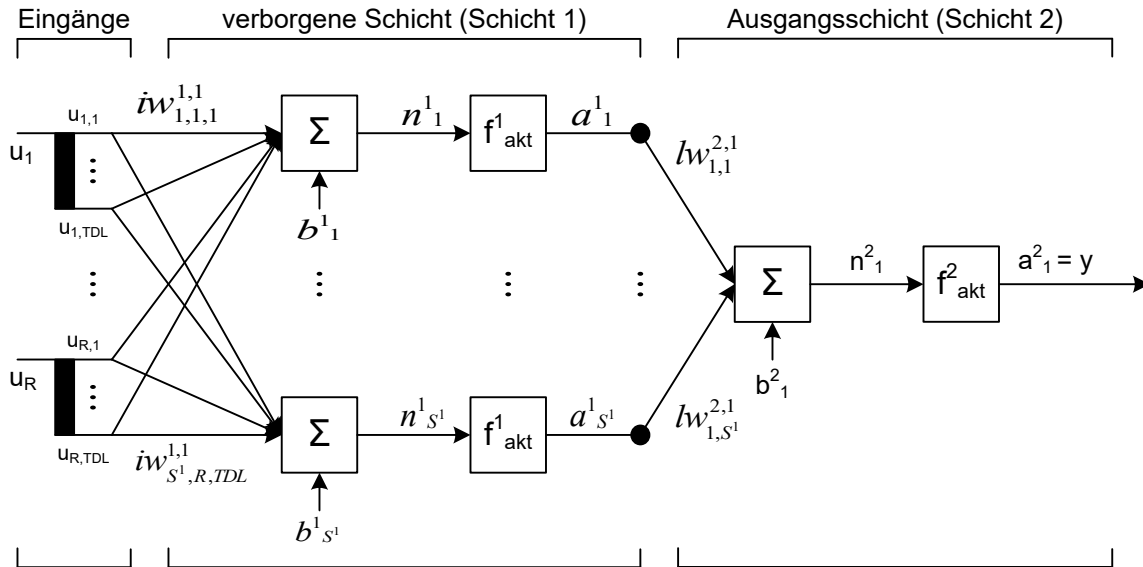


Abbildung 5.5: Blockschaltbild eines zweischichtigen NIO-Netzes mit  $R$  Eingangsgrößen,  $S^1$  Neuronen, einer Ausgangsgröße und TDL zurückliegenden Messwerten

Mit den Tapped Delay Lines ändert sich die Dimension der Wichtungsmatrix  $IW$ . Zur eindeutigen Zuordnung wird ein weiterer Index hinzugefügt. Der dritte tiefgestellte Index gibt an, welcher der zurückliegenden Messwerte gewichtet wird. Die TDL wirken sich nur auf die Berechnung der Neuronen-Eingangswerte  $n^1$  aus. Für das erste Neuron folgt

$$n^1_1 = (iw_{1,1,1}^{1,1} \cdot u_{1,1} + \dots + iw_{1,1,TDL}^{1,1} \cdot u_{1,TDL}) + \dots + (iw_{1,R,1}^{1,1} \cdot u_{R,1} + \dots + iw_{1,R,TDL}^{1,1} \cdot u_{R,TDL}) + b^1_{1,1} \quad (5.9)$$

$$n^1_1 = \left( \sum_{i=1}^R \left( \sum_{d=1}^{TDL} iw_{1,i,d}^{1,1} \cdot u_{i,d} \right) \right) + b^1_{1,1} \quad (5.10)$$

Für die Ausgangsgröße des in Abbildung 5.5 dargestellten Netzes folgt schließlich

$$y = \left[ \sum_{j=1}^{S^1} lw_{j,1}^{2,1} \cdot \left( \sum_{i=1}^R \left( \sum_{d=1}^{TDL} iw_{j,i,d}^{1,1} \cdot u_{i,d} \right) \right) + b^1_{j,1} \right] + b^2_{1,1} \quad (5.11)$$

Zur NARX-Struktur müssen die Ausgangsgrößen zurückgeführt werden (Abbildung 5.6). Diese wirken sich ebenfalls nur auf die Gleichungen der Neuronen-Eingangswerte  $n^1$  aus. Die zurückgeführten Ausgangswerte werden mit  $lw$  verstärkt. Der Gleichung (5.10) wird ein entsprechender Term für die zurückgeführten Ausgangsgrößen hinzugefügt.

$$n^1_1 = \left( \sum_{i=1}^R \left( \sum_{d=1}^{TDL} iw_{1,i,d}^{1,1} \cdot u_{i,d} \right) \right) + \left( \sum_{d=1}^{TDL} lw_{1,1,d}^{1,2} \cdot y_{1,d} \right) + b^1_1. \quad (5.12)$$

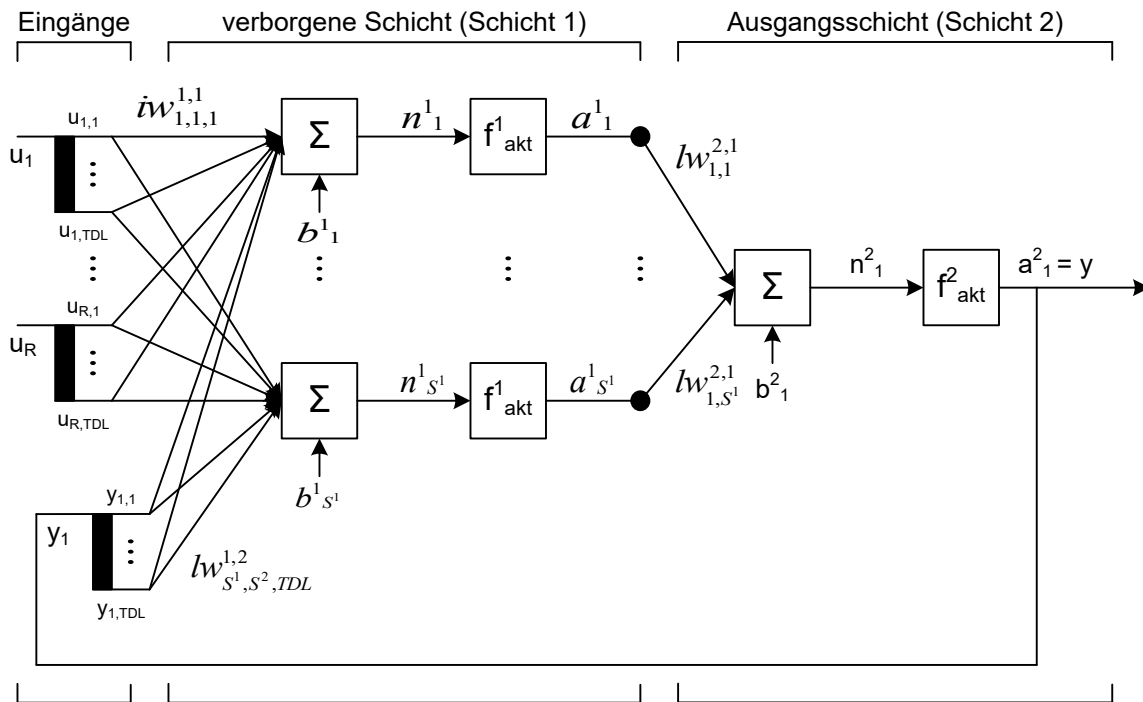


Abbildung 5.6: Vereinfachtes Blockschaltbild eines zweischichtigen NARX-Netzes mit  $R$  Eingangsgrößen,  $S^1$  Neuronen, einer Ausgangsgröße und TDL zurückliegenden Messwerten

Für ein zweischichtiges NARX-Netz mit  $R$  Eingangsgrößen, einer Ausgangsgröße,  $S^1$  verborgenen Neuronen, einem betrachteten zurückliegenden Zeithorizont TDL und linearer Aktivierungsfunktion ergibt sich die zukünftige Ausgangsgröße nach folgender Formel.

$$y = \left( \sum_{j=1}^{S^1} lw_{1,j}^{2,1} \cdot \left[ \left( \sum_{i=1}^R \left( \sum_{d=1}^{TDL} iw_{j,i,d}^{1,1} \cdot u_{i,d} \right) \right) + \left( \sum_{d=1}^{TDL} lw_{j,1,d}^{1,2} \cdot y_{1,d} \right) + b^1_j \right] \right) + b^2_1 \quad (5.13)$$



Mit einer zeitabhängigen Notation ergibt sich die Gleichung zur OSP eines NARX-Netzes zu

$$\text{NARX: } y(t) = \left( \sum_{j=1}^{S^1} lw_{1,j}^{2,1} \cdot \left[ \left( \sum_{i=1}^R \left( \sum_{d=1}^{TDL} iw_{j,i,d}^{1,1} \cdot u_i(t-d) \right) \right) + \left( \sum_{d=1}^{TDL} lw_{j,1,d}^{1,2} \cdot y(t-d) \right) + b^1_j \right] \right) + b^2_1. \quad (5.14)$$

Die Abbildung 5.7 veranschaulicht die Gleichung (5.14) und das Grundprinzip der OSP an einem vereinfachten Beispiel für ein SISO-NARX-Netz. Zur Berechnung der Ausgangsgröße  $y(t+k)$  mit  $k=0$  benötigt das Netz die letzten zehn Werte der Ein- und Ausgangsgrößen, die nach Gleichung (5.14) entsprechend gewichtet werden.

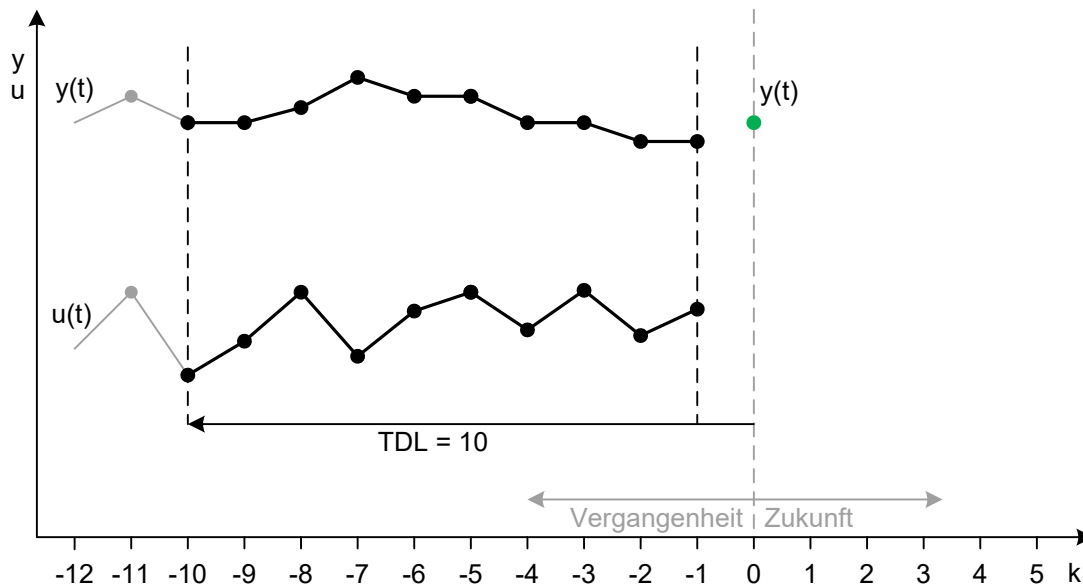


Abbildung 5.7: Grundprinzip der OSP für ein NARX-Netz mit zehn TDL

Die Gleichungen für NIO- oder NAR-Netze lauten wie folgt.

$$\text{NAR: } y(t) = \left( \sum_{j=1}^{S^1} lw_{1,j}^{2,1} \cdot \left[ \left( \sum_{d=1}^{TDL} lw_{j,1,d}^{1,2} \cdot y(t-d) \right) + b^1_j \right] \right) + b^2_1 \quad (5.15)$$

$$\text{NIO: } y(t) = \left( \sum_{j=1}^{S^1} lw_{1,j}^{2,1} \cdot \left[ \left( \sum_{i=1}^R \left( \sum_{d=1}^{TDL} iw_{j,i,d}^{1,1} \cdot u_i(t-d) \right) \right) + b^1_j \right] \right) + b^2_1 \quad (5.16)$$

Im Anhang F sind die ausführlichen Blockschaltbilder und Gleichungen der fünf evaluierten Netze ergänzend dargestellt.

### 5.2.2.2 Multi Step Prediction

Zum Aufstellen der Gleichungen zur MSP ist das Verschieben der Zeithorizonte notwendig, was eine Trennung in vergangene und in der Zukunft liegende Bereiche erfordert. Eine weitere Differenzierung erfolgt durch die Tatsache, dass der Stellhorizont  $N_u$ , also der Zeitraum, für den die zukünftigen Stellgrößen optimiert werden sollen, in der Regel kleiner als der Prädiktionshorizont  $N_2$  gewählt wird. Alle zukünftigen Stellgrößen, die über den Stellhorizont hinausgehen, sind konstant. In Anlehnung an Abbildung 5.7 veranschaulicht Abbildung 5.8 das Grundprinzip der MSP an einem vereinfachten Beispiel mit  $k = 5$ . Der Stellhorizont wurde in diesem Fall auf  $N_u = 2$  gesetzt.

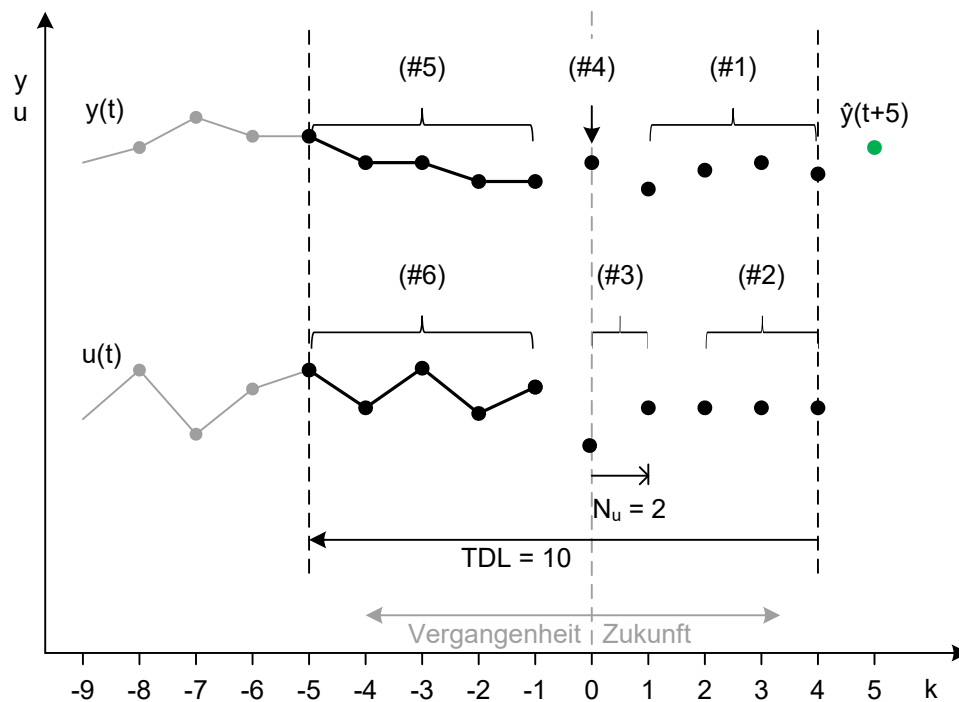


Abbildung 5.8: Grundprinzip der Multi Step Prediction am Beispiel eines NARX-Netzes mit  $TDL = 10$  und  $N_u = 2$

Für die Berechnung der Regelgröße  $\hat{y}(t+5)$  benötigt das Netz Ein- und Ausgangsgrößen, die zum Teil noch in der Vergangenheit und bereits zum Teil in der Zukunft liegen. Abhängig vom aktuellen Zeitpunkt der Prädiktion, vom TDL-Wert und vom Stellhorizont ergeben sich verschiedene Blöcke.

Die Einteilung erfolgt allgemein in

- (#1) alle bis zum Zeitpunkt  $(t + k)$  bereits prädizierten Regelgrößen,
- (#2) alle über den Stellhorizont  $N_u$  hinausgehenden konstanten Stellgrößen,
- (#3) alle  $N_u$  zu optimierenden Stellgrößen (einschließlich  $u(t)$ ),
- (#4) die zum Zeitpunkt  $(t - 1)$  vorhergesagte Regelgröße  $\hat{y}(t)$ ,
- (#5) alle restlichen  $(TDL - k)$  tatsächlich gemessenen Regelgrößen und
- (#6) alle restlichen  $(TDL - k)$  tatsächlich ausgegebenen Stellgrößen.

Zur Berechnung der Neuronen-Eingangswerte  $n^1$  wird die Gleichung (5.12) in die sechs Summen zerlegt. Dabei wird mit

- $k$  der aktuelle Zeitpunkt im Prädiktionszeitraum,
- $TDL$  der für die Berechnung notwendige zurückliegende Zeithorizont und
- $N_u$  der Stellhorizont

bezeichnet. Für die Berechnung der von  $k$  abhängigen Neuronen-Eingangswerte  $n^1$  ergibt sich in Anlehnung an Abbildung 5.8 folgende Gleichung.

$$\begin{aligned}
 \tilde{n}^1_j(k) = & \sum_{d=1}^{\min(k-1, TDL)} lw_{j,1,d}^{1,2} \cdot \hat{y}(t+k-d) & \text{(#1)} & \quad (5.17) \\
 & + \sum_{d=1}^{\min(k-N_u+1, TDL)} iw_{j,1,d}^{1,1} \cdot u(t+N_u-1) & \text{(#2)} & \\
 & + \sum_{d=k-N_u+1}^{\min(k-1, TDL-1)} iw_{j,1,d+1}^{1,1} \cdot u(t+k-1+d) & \text{(#3)} & \\
 & + \sum_{d=k}^{\min(k, TDL)} lw_{j,1,d}^{1,2} \cdot \hat{y}(t+k-d) & \text{(#4)} & \\
 & + \sum_{d=k+1}^{TDL} lw_{j,1,d}^{1,2} \cdot y(t+k-d) & \text{(#5)} & \\
 & + \sum_{d=k}^{TDL-1} iw_{j,1,d}^{1,1} \cdot u(t+k-1-d) & \text{(#6)} & \\
 & + b^1_j
 \end{aligned}$$

Die Tilde kennzeichnet, dass der berechnete Wert zum Teil aus zukünftigen und vergangenen Komponenten besteht. Alle prädizierten Regelgrößen sind mit dem Zirkumflex markiert. Die in Gleichung (5.17) verwendeten Indizes sind

- $j$  für die Anzahl der verborgenen Neuronen mit  $1 \leq j \leq S^1$  und
- $d$  für den jeweiligen Zeitpunkt im *TDL*-Zeitraum.

Die Gleichung zur Multi Step Prediction eines NARX-Netz folgt schließlich zu

$$\hat{y}(t+k) = \left( \sum_{j=1}^{S^1} lw_{1,j}^{2,1} \cdot \tilde{n}^1_j(k) \right) + b^2_1. \quad (5.18)$$

Für den Fall, dass in der verborgenen Schicht die Aktivierungsfunktion *tansig* verwendet wird, ändert sich die Gleichung (5.18) zu

$$\hat{y}(t+k) = \left( \sum_{j=1}^{S^1} lw_{1,j}^{2,1} \cdot \tanh(\tilde{n}^1_j(k)) \right) + b^2_1 \quad (5.19)$$

Die Gleichungen für NIO- oder NAR-Netze lassen sich durch Reduzieren der nicht vorhandenen Strukturelemente aufstellen. Für das NIO-Netz 4 entfällt die Summe (#5) in Gleichung (5.17). Zusammengefasst kann mit den Gleichungen (5.17) und (5.18) die Regelgröße für einen beliebigen Zeitpunkt  $k$  im Prädiktionszeitraum berechnet werden.

### 5.2.3 Gütekriterium

Das Gütekriterium hat entscheidenden Einfluss auf das Regelkreisverhalten. Es steht die Frage im Vordergrund, welchen Zweck die Regelung erfüllen soll. Zur Reduzierung der Betriebskosten und zur Erhöhung der Wirtschaftlichkeit kommen Kriterien infrage, die den Energiehaushalt bewerten und Sollwerte für unterlagerte Regelkreise vorgeben. Der Fokus wird auf eine übergeordnete Optimierung des Gesamtprozesses gelegt. Bei kritischen und sensiblen Prozessen liegt der Schwerpunkt eher in einer gesicherten Prozessführung, bei der u. a. Regel- und Stellgrößen definierte Grenzen aus Sicherheitsgründen nicht verlassen dürfen oder die Einhaltung vorgegebener Sollwerte aus Qualitätsgründen unabdingbar ist. Mit der Auswahl des Gütekriteriums geht auch die Anzahl zusätzlicher Reglerparameter einher. Mit zusätzlichen Wichtungsfaktoren lassen sich prozessspezifische Anforderungen bereits im Gütekriterium berücksichtigen.

Im Rahmen dieser Arbeit steht nicht die übergeordnete Optimierung des Gesamtprozesses der Flachglasherstellung im Vordergrund, sondern die konzeptionelle Betrachtung einer prädiktiven Regelung auf Basis der evaluierten KNN. Aus diesem Grund wird ein Kriterium gewählt, das vornehmlich prozessnahe Kenngrößen bewertet, die unmittelbar von den KNN bereit gestellt werden. Dazu gehören zukünftige Regel- und Stellgrößen und deren Änderungen. Ein oftmals in der Literatur beschriebener und aus der klassischen Regelungstechnik bekannter Ansatz ist die Minimierung der quadratischen Abweichung zwischen einem gegebenen Sollwertverlauf und den Regelgrößen im Prädiktionszeitraum [67, 70, 73, 75 - 77]. Zusätzlich kann ein Term zur Bewertung des Stellaufwandes hinzugefügt werden, mit dem die Änderungen der zukünftigen Stellgrößen so gering wie möglich gehalten werden sollen. Damit kann ein sanftes und stabilisierend wirkendes Stellverhalten realisiert werden. Das Gütekriterium  $J$ , welches von der Zeit  $t$  und dem Stellgrößenvektor  $U(t)$  abhängt, lässt sich formal zu

$$J(t, U(t)) = \sum_{k=N_1}^{N_2} (r(t+k) - \hat{y}(t+k))^2 + \rho \sum_{k=1}^{N_u} (\Delta u(t+k-1))^2 \quad (5.20)$$

schreiben. Darin sind  $r(t)$  die zukünftigen Sollwerte,  $N_u$  das Ende des Stellhorizontes,  $N_1$  der Beginn und  $N_2$  das Ende des Prädiktionshorizontes. Der Wichtungsfaktor  $\rho$  bewertet den Stellaufwand. Die Stellgrößenänderungen ergeben sich nach

$$\Delta u(t+k-1) = u(t+k-1) - u(t+k-2).$$

Für alle Stellgrößen, die über den Stellhorizont  $N_u$  hinausgehen, gilt  $\Delta u = 0$ . Nach Transformation des Gütekriteriums in eine äquivalente Matrizenschreibweise folgt

$$J(t, U(t)) = \left[ (R(t) - \hat{Y}(t))^T \cdot (R(t) - \hat{Y}(t)) \right] + \rho [\Delta U^T(t) \cdot \Delta U(t)] \quad (5.21)$$

bzw.

$$J(t, U(t)) = E^T(t) \cdot E(t) + \rho [\Delta U^T(t) \cdot \Delta U(t)] . \quad (5.22)$$

Die Matrizen und Vektoren sind wie folgt aufgebaut.

- Zukünftige Sollwerte  $R(t) = \begin{bmatrix} r(t + N_1) \\ \vdots \\ r(t + N_2) \end{bmatrix}$
- Prädizierte Regelgrößen  $\hat{Y}(t) = \begin{bmatrix} \hat{y}(t + N_1 | t) \\ \vdots \\ \hat{y}(t + N_2 | t) \end{bmatrix}$
- Zukünftige Stellgrößen  $U(t) = \begin{bmatrix} u(t) \\ \vdots \\ u(t + N_u - 1) \end{bmatrix}$
- Zukünftige Stellgrößenänderungen  $\Delta U(t) = \begin{bmatrix} \Delta u(t) \\ \vdots \\ \Delta u(t + N_u - 1) \end{bmatrix}$
- Zukünftige Regelabweichungen  $E(t) = \begin{bmatrix} r(t + N_1) - \hat{y}(t + N_1 | t) \\ \vdots \\ r(t + N_2) - \hat{y}(t + N_2 | t) \end{bmatrix} = \begin{bmatrix} e(t + N_1 | t) \\ \vdots \\ e(t + N_2 | t) \end{bmatrix}$

Die Schreibweise  $\hat{y}(t + k | t)$  bedeutet die zum Zeitpunkt  $t$  berechnete Vorhersage der Regelgröße für den Zeitpunkt  $(t + k)$ . Mit dem in Gleichung (5.22) aufgestellten Gütekriterium kann zusammengefasst die quadratische Regelabweichung im Prädiktionszeitraum und gleichzeitig der quadratische Stellaufwand minimiert werden. Die prädizierten Regelgrößen  $\hat{Y}(t)$  werden vom Prädiktor bereitgestellt. Der Sollwertverlauf  $R(t)$  ist gegeben.

## 5.2.4 Optimierung nach dem Levenberg-Marquardt-Algorithmus

### 5.2.4.1 Entscheidungsgrundlage

Eine Übersicht möglicher Optimierungsverfahren zur Lösung unrestringierter Optimierungsprobleme für quadratische Gütefunktionen bieten u. a. [78, 79]. Für die numerische Lösung des vorliegenden nichtlinearen Optimierungsproblems wird der nach Kenneth Levenberg und Donald Marquardt benannte Algorithmus gewählt. Weiterführende Informationen, insbesondere zum LM-Algorithmus, finden sich u. a. in [70, 80, 81].

Es sprechen mehrere Punkte für den Levenberg-Marquardt-Algorithmus.

- Das zu minimierende Gütekriterium (5.20) ist eine quadratische Funktion, weshalb iterative Suchmethoden zweiter Ordnung vorzuziehen sind. Neben dem LM-Algorithmus kommen noch Newton-basierte Algorithmen (z. B. Gauss-Newton oder Quasi-Newton) infrage.
- Der Algorithmus ist eine Kombination aus dem klassischen Gradientenverfahren und dem Gauss-Newton-Verfahren. Die Suchrichtung wird über den Parameter  $\lambda$  gesteuert. Bei großem Abstand zum Minimum entspricht die Suchrichtung der des Gradientenverfahrens und bei kleinem Abstand der des Gauss-Newton-Verfahrens. Die Kombination verbindet somit die Vorteile beider Verfahren. Mit dem Gradientenansatz kann schnell in die Richtung des Optimums geschritten werden, in dessen Nähe anschließend die Gütefunktion durch eine quadratische Näherung ersetzt wird, die explizit nach dem Minimum gelöst werden kann.
- Mit dem LM-Parameter  $\lambda$  wird gleichzeitig die Schrittweite und die Suchrichtung gesteuert. Damit ist kein zusätzlicher Algorithmus zur Schrittweitensteuerung notwendig, wie es z. B. bei einem Quasi-Newton-Ansatz der Fall wäre.
- Der LM-Algorithmus lieferte bereits beim Training der KNN die beste Performance (vgl. Abschnitt 3.2.2). Das Training stellt ebenfalls eine nichtlineare Optimierungsaufgabe dar, bei der der quadratische Fehler (MSE) minimiert wird. Besonders die schnelle Konvergenz und die hohe Genauigkeit im Vergleich zu den anderen Optimierungsverfahren sind ausschlaggebend.

#### 5.2.4.2 Herleitung des Regelgesetzes nach Levenberg-Marquardt

Ausgangspunkt der folgenden Gleichungen sind die detaillierten Ausführungen in [70], die auf den vorliegenden Anwendungsfall übertragen und ergänzt werden. Zur Berechnung der zukünftigen Stellgrößenfolge  $U(t)$  ist eine iterative Annäherung in der Form

$$U^{(i+1)} = U^{(i)} + f^{(i)} \quad (5.23)$$

notwendig, wobei  $f$  die Suchrichtung der  $i$ -ten Iteration ist. Der Unterschied der einzelnen Optimierungsmethoden besteht hauptsächlich in der Berechnung der Suchrichtung  $f$ . Beim klassischen Gradientenverfahren entspricht die Suchrichtung im einfachsten Fall dem negativen Gradienten  $G$ . Beim Newton-Verfahren wird eine Taylor-Approximation an der aktuellen Suchstelle der Gütefunktion durchgeführt. Neben dem Gradienten wird die zweite

partielle Ableitung der Gütefunktion (Hesse-Matrix  $H$ ) benötigt. Die Konvergenz mit dem Newton-Verfahren ist deutlich schneller, jedoch erfordert die Berechnung der Hesse-Matrix großen Aufwand. Beim Gauss-Newton-Verfahren wird versucht, bei jedem Iterationsschritt die Residuen mit der Methode der kleinsten Quadrate zu minimieren. Für die Lösung wird nicht die exakte Hesse-Matrix benötigt, sondern nur eine reduzierte Form, die hier mit  $R$  bezeichnet wird. Für den LM-Algorithmus ergibt sich die Suchrichtung letztlich aus einer Kombination der Gleichungen (5.24) und (5.26). Für  $\lambda \rightarrow 0$  strebt die LM-Suchrichtung zur Suchrichtung des Gauss-Newton-Verfahrens.

$$\text{Gradientenverfahren:} \quad f^{(i)} = -G(U^{(i)}) \quad (5.24)$$

$$\text{Newton-Verfahren:} \quad H(U^{(i)}) \cdot f^{(i)} = -G(U^{(i)}) \quad (5.25)$$

$$\text{Gauss-Newton-Verfahren:} \quad R(U^{(i)}) \cdot f^{(i)} = -G(U^{(i)}) \quad (5.26)$$

$$\text{Levenberg-Marquardt-Verfahren:} \quad [R(U^{(i)}) + \lambda^{(i)} \cdot I] \cdot f^{(i)} = -G(U^{(i)}) \quad (5.27)$$

Die hier gestellte Optimierungsaufgabe besteht darin, zukünftige Stellgrößen über einen Stellhorizont zu berechnen, mit denen die quadratischen Regelabweichungen im Prädiktionshorizont minimal werden. Aufgrund von Modellunsicherheiten des Netzes können die prädizierten Regelgrößen zu größeren Residuen führen, was sich beim Gauss-Newton-Verfahren negativ auf die Konvergenzgeschwindigkeit auswirken würde. In [70] wird aus diesem Grund ein modifizierter LM-Algorithmus vorgestellt, der anstatt der Gauss-Newton-Hesse-Matrix  $R$  die Hesse-Matrix  $H$  des vollen Newton-Verfahrens verwendet. Der Nachteil besteht in der aufwendigen Berechnung dieser Matrix, der Vorteil in der schnelleren Konvergenz. Die Suchrichtung des Newton-basierten LM-Algorithmus ergibt sich somit zu

$$\text{Newton-basierter LM:} \quad [H(U^{(i)}) + \lambda^{(i)} \cdot I] \cdot f^{(i)} = -G(U^{(i)}). \quad (5.28)$$

Zur Berechnung der Suchrichtung sind der Gradientenvektor und die Hesse-Matrix notwendig. Beide Größen hängen von den zukünftigen Stellgrößen  $U(t)$  ab.

$$\begin{aligned} G(U^{(i)}(t)) &= \frac{\partial J(t, U(t))}{\partial U(t)} \\ G(U^{(i)}(t)) &= -2 \frac{\partial \hat{Y}(U(t))^T}{\partial U(t)} E(t) + 2\rho \frac{\partial \Delta U(U(t))^T}{\partial U(t)} \Delta U(U(t)) \end{aligned} \quad (5.29)$$



$$H(U^{(i)}(t)) = \frac{\partial^2 J(t, U(t))}{\partial U(t)^2} = \frac{\partial G(U^{(i)}(t))}{\partial U(t)}$$

$$H(U^{(i)}(t)) = -2 \frac{\partial}{\partial U(t)} \left( \frac{\partial \hat{Y}(U(t))^T}{\partial U(t)} E(t) \right) + 2\rho \frac{\partial}{\partial U(t)} \left( \frac{\partial \Delta U(U(t))^T}{\partial U(t)} \Delta U(U(t)) \right) \quad (5.30)$$

$$H(U^{(i)}(t)) = -2 \left( \frac{\partial^2 \hat{Y}(U(t))^T}{\partial U(t)^2} E(t) - \frac{\partial \hat{Y}(U(t))^T}{\partial U(t)} \frac{\partial \hat{Y}(U(t))}{\partial U(t)} \right) +$$

$$2\rho \left( \frac{\partial^2 \Delta U(U(t))^T}{\partial U(t)^2} \Delta U(U(t)) + \frac{\partial \Delta U(U(t))^T}{\partial U(t)} \frac{\partial \Delta U(U(t))}{\partial U(t)} \right) \quad (5.31)$$

Zur Berechnung des Gradientenvektors nach Gleichung (5.29) müssen die partiellen Ableitungen  $\partial \Delta U(U(t))^T / \partial U(t)$  und  $\partial \hat{Y}(U(t))^T / \partial U(t)$  ermittelt werden.

$$\frac{\partial \Delta U(U(t))}{\partial U(t)} = \begin{bmatrix} \frac{\partial \Delta u(t)}{\partial u(t)} & \dots & \frac{\partial \Delta u(t)}{\partial u(t + N_u - 1)} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Delta u(t + N_u - 1)}{\partial u(t)} & \dots & \frac{\partial \Delta u(t + N_u - 1)}{\partial u(t + N_u - 1)} \end{bmatrix} \quad (5.32)$$

$$\frac{\partial \hat{Y}(U(t))}{\partial U(t)} = \begin{bmatrix} \frac{\partial \hat{y}(t + N_1)}{\partial u(t)} & \dots & \frac{\partial \hat{y}(t + N_1)}{\partial u(t + N_u - 1)} \\ \vdots & \ddots & \vdots \\ \frac{\partial \hat{y}(t + N_2)}{\partial u(t)} & \dots & \frac{\partial \hat{y}(t + N_2)}{\partial u(t + N_u - 1)} \end{bmatrix} \quad (5.33)$$

Zur Wahrung der nötigen Dimensionen sind die beiden Matrizen noch zu transponieren. Die Matrix (5.32) kann mit wenig Aufwand berechnet werden. Bis auf die partiellen Ableitungen  $\partial \Delta u(t + l) / \partial u(t + l) = 1$  und  $\partial \Delta u(t + l + 1) / \partial u(t + l) = -1$  für  $0 \leq l \leq N_u - 1$  sind die restlichen Elemente per Definition Null. In der Matrix (5.33) sind die partiellen Ableitungen der prädizierten Regelgrößen nach den zu optimierenden Stellgrößen enthalten. Die zukünftigen Regelgrößen  $\hat{y}(t + k)$  lassen sich nach den bereits hergeleiteten Gleichungen (5.18) bzw. (5.19) berechnen. Die Komponenten der Matrix (5.33) setzen sich folglich aus der Ableitung der Gleichung (5.18) zusammen. Mit linearer Aktivierungsfunktion folgt

$$\frac{\partial \hat{y}(t + k)}{\partial u(t + l)} = \sum_{j=1}^{s^l} lw_{1,j}^{2,l} \cdot \frac{\partial \tilde{n}_j^1(k)}{\partial u(t + l)} \quad \text{mit} \quad \begin{matrix} k \in [d, N_2] \\ l \in [0, \min(k - 1, N_u - 1)] \end{matrix} \quad (5.34)$$

Die partiellen Ableitungen  $\partial \tilde{n}_j^1(k) / \partial u(t+l)$  ergeben sich zu

(5.35)

$$\frac{\partial \tilde{n}_j^1(k)}{\partial u(t+l)} = \begin{cases} \sum_{d=1}^{\min(k-1-l, TDL)} l w_{j,1,d}^{1,2} \frac{\partial \hat{y}(t+k-d)}{\partial u(t+l)} + \sum_{d=1}^{\min(k-N_u-1, TDL)} i w_{j,1,d}^{1,1} & \text{für } l = N_u - 1 \\ \sum_{d=1}^{\min(k-1-l, TDL)} l w_{j,1,d}^{1,2} \frac{\partial \hat{y}(t+k-d)}{\partial u(t+l)} + i w_{j,1,k-l}^{1,1} & \text{für } \max(0, k - TDL) \leq l \leq N_u - 2 \\ \sum_{d=1}^{\min(k-1-l, TDL)} l w_{j,1,d}^{1,2} \frac{\partial \hat{y}(t+k-d)}{\partial u(t+l)} & \text{für } 0 \leq l \leq \max(0, k - TDL) \end{cases}$$

In Anlehnung an Abbildung 5.8 veranschaulicht Abbildung 5.9 die Zusammensetzung der in Gleichung (5.35) enthaltenen partiellen Ableitungen. Es sind alle bereits prädizierten Regelgrößen  $\hat{y}(t+k)$  nach allen  $N_u$  zu optimierenden Stellgrößen partiell abzuleiten. Liegt der aktuelle Zeitpunkt  $k$  innerhalb vom Stellhorizont  $N_u$ , so sind die bis dato prädizierten Regelgrößen nur nach den Stellgrößen bis zum Zeitpunkt  $(t+k-1)$  abzuleiten.

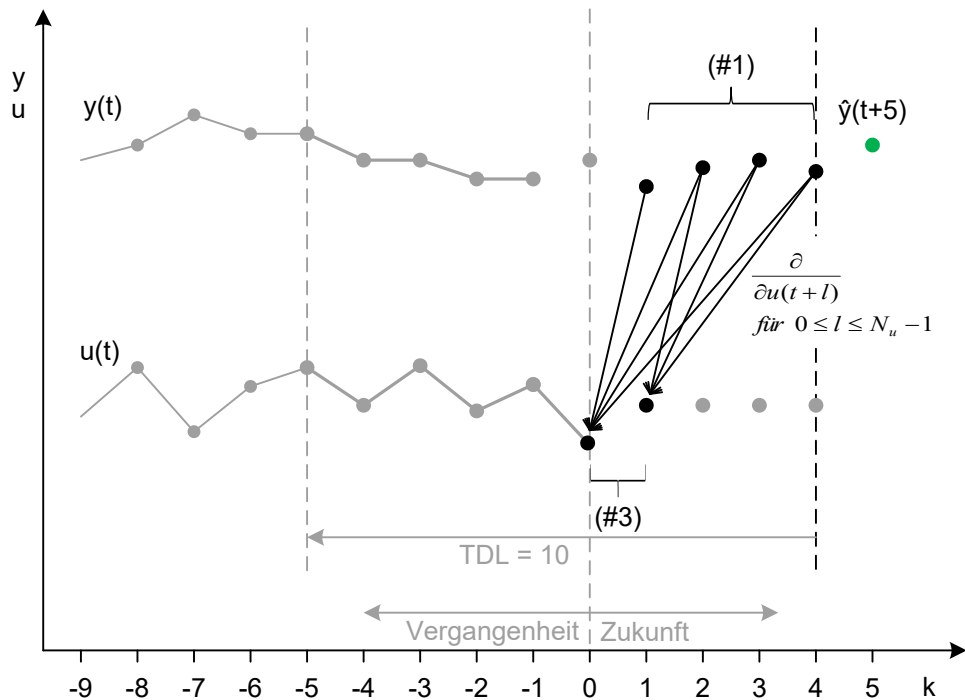


Abbildung 5.9: Beispiel der zu berechnenden partiellen Ableitungen nach Gleichung (5.35)

Zur Berechnung der Hesse-Matrix (5.31) sind weitere partielle Ableitungen notwendig. Die Berechnung der doppelten partiellen Ableitung des Regelgrößenvektors  $\hat{Y}(U(t))$  nach den

Stellgrößenvektor  $U(t)$ , multipliziert mit den zukünftigen Regelabweichungen  $E(t)$  erfordert großen Aufwand und lässt sich formal nach

$$\frac{\partial^2 \hat{Y}(U(t))^T}{\partial (U(t))^2} E(t) = \begin{bmatrix} \frac{\partial^2 \hat{Y}(U(t))^T}{\partial u(t)^2} \cdot E(t) & \cdots & \frac{\partial^2 \hat{Y}(U(t))^T}{\partial u(t) \partial u(t + N_u - 1)} \cdot E(t) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 \hat{Y}(U(t))^T}{\partial u(t) \partial u(t + N_u - 1)} \cdot E(t) & \cdots & \frac{\partial^2 \hat{Y}(U(t))^T}{\partial u(t + N_u - 1)^2} \cdot E(t) \end{bmatrix} \quad (5.36)$$

berechnen. Für jede einzelne zu berechnende partielle Ableitung zweiter Ordnung  $\partial^2 \hat{y}(t+k) / [\partial u(t+l) \partial u(t+p)]$  folgt nach erneuter Ableitung der Gleichung (5.34)

$$\frac{\partial^2 \hat{y}(t+k)}{\partial u(t+l) \partial u(t+p)} = \sum_{j=1}^{S^l} l w_{1,j}^{2,1} \cdot \frac{\partial^2 \tilde{n}_j^1(k)}{\partial u(t+l) \partial u(t+p)} \quad (5.37)$$

mit

$$\frac{\partial^2 \tilde{n}_j^1(k)}{\partial u(t+l) \partial u(t+p)} = \sum_{d=1}^{\min(k-l-1, TDL)} l w_{j,1,d}^{1,2} \frac{\partial \hat{y}(t+k-d)}{\partial u(t+l) \partial u(t+p)} \quad (5.38)$$

für alle  $k \in [d, N_2]$ ,  $l \in [0, \min(k-1, N_u-1)]$  und  $p \in [0, l]$ . Insbesondere bei der Berechnung der Hesse-Matrix stellt sich die Nutzung der linearen Aktivierungsfunktion als Vorteil heraus. Der Berechnungsaufwand ist deutlich geringer. Zum Vergleich sei die Gleichung bei Verwendung der *tansig*-Aktivierungsfunktion noch aufgeführt.

$$\begin{aligned} \frac{\partial^2 \hat{y}(t+k)}{\partial u(t+l) \partial u(t+p)} = \sum_{j=1}^{S^l} l w_{1,j}^{2,1} \cdot & \left( \left[ -2 \tanh(\tilde{n}_j^1(k)) \right] \cdot \left( 1 - \tanh^2(\tilde{n}_j^1(k)) \right) \cdot \frac{\partial \tilde{n}_j^1(k)}{\partial u(t+l)} \cdot \frac{\partial \tilde{n}_j^1(k)}{\partial u(t+l)} \right. \\ & \left. + \left( 1 - \tanh^2(\tilde{n}_j^1(k)) \right) \frac{\partial}{\partial u(t+p)} \left( \frac{\partial \tilde{n}_j^1(k)}{\partial u(t+l)} \right) \right) \end{aligned} \quad (5.39)$$

Die partielle Ableitung  $\partial^2 \Delta U(U(t))^T / \partial U(t)^2$ , also die erneute Ableitung von der Matrix (5.32), wird per Definition Null, weshalb sich die Gleichung (5.31) zur Berechnung der Hesse-Matrix in diesem Fall reduziert. Die restlichen Produkte in Gleichung (5.31) können mit den bisher aufgestellten Formeln vollständig berechnet werden.

Mit folgender Formel wird anschließend ermittelt, inwieweit sich das Gütekriterium verbessert hat [70]. Abhängig vom Verhältnis  $r^{(i)}$  wird der Parameter  $\lambda$  erneut angepasst (siehe auch Abbildung 5.10).

$$r^{(i)} = 2 \frac{J(U^{(i)}) - J(U^{(i)} + f^{(i)})}{\lambda \cdot (f^{(i)})^T \cdot f^{(i)} - (f^{(i)})^T \cdot G(U^{(i)})} \quad (5.40)$$

#### 5.2.4.3 Cholesky-Zerlegung

In Abschnitt 5.2.4.2 wurden die Gleichungen zur Berechnung der Suchrichtung, des Gradientenvektors und der Hesse-Matrix hergeleitet. Eine notwendige Bedingung, um die neue Suchrichtung bestimmen zu können, ist die positive Definitheit der Matrix  $[H(U^{(i)}) + \lambda^{(i)}I]$ . Diese Überprüfung kann mit der Cholesky-Zerlegung erfolgen, die nur erfolgreich ist, wenn die zu untersuchende symmetrische Matrix positiv bestimmt ist. Bei der Cholesky-Zerlegung wird die zu testende Matrix in ein Produkt aus einer unteren Dreiecksmatrix  $L$  und der Transponierten dieser Dreiecksmatrix zerlegt.

$$[H(U^{(i)}) + \lambda^{(i)}I] = L \cdot L^T = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{p1} & l_{p2} & \cdots & l_{pp} \end{bmatrix} \cdot \begin{bmatrix} l_{11} & l_{21} & \cdots & l_{p1} \\ 0 & l_{22} & \cdots & l_{p2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & l_{pp} \end{bmatrix} \quad (5.41)$$

Das resultierende Gleichungssystem wird nach den Cholesky-Faktoren  $l_{xy}$  umgestellt. Kann ein Faktor nicht berechnet werden, z. B. aufgrund eines negativen Wurzelausdrucks, ist die Matrix nicht positiv bestimmt. In diesem Fall wird der LM-Parameter  $\lambda$  entsprechend nach dem Algorithmus von Levenberg-Marquardt angepasst und die Cholesky-Zerlegung so oft wiederholt, bis die positive Bestimmtheit hergestellt ist. Nach erfolgreicher Zerlegung kann die neue Suchrichtung berechnet werden. Dabei wird das Gaußsche Eliminationsverfahren mit Vorwärts- und Rückwärtseinsetzen angewendet.

$$[L \cdot L^T] \cdot f^{(i)} = -G$$

$$\text{Vorwärtseinsetzen: } L \cdot x = -G \quad (5.42)$$

$$\text{Rückwärtseinsetzen: } L^T \cdot f^{(i)} = x \quad (5.43)$$

Die neue Suchrichtung  $f^{(i)}$  wird durch Umstellen der Gleichung (5.43) berechnet.

#### 5.2.4.4 Zusammenfassung und Ablaufplan

In den vorherigen Abschnitten wurden die Gleichungen zur Prädiktion der Regelgrößen, zur Berechnung des Gütekriteriums, des Gradientenvektors und der Hesse-Matrix aufgestellt. Der gesamte Ablaufplan und die dafür notwendigen Rechenschritte zur Berechnung der zukünftigen optimalen Stellgrößenfolge  $U(t)$  sind in Abbildung 5.10 dargestellt. Der Optimierungsalgorithmus wird zu jedem Abtastzeitpunkt durchgeführt und von der Stellgrößenfolge  $U(t)$  wird lediglich das erste Element  $u(t)$  an den Prozess ausgegeben. Zum nächsten Abtastzeitpunkt wird die Optimierungsaufgabe mit neuen eingelesenen Messwerten erneut gelöst. Die Zeitfenster verschieben sich fortwährend.

Die Suchrichtung wird durch den LM-Parameter  $\lambda$  gesteuert. Dieser Parameter wird in Kombination mit der Cholesky-Zerlegung so angepasst, dass die neue Suchrichtung zwangsläufig zu einer Minimierung der Gütefunktion führt.

In Abbildung 5.10 sind zwei übliche Abbruchbedingungen integriert, die den Berechnungsaufwand und die Qualität der Lösung beeinflussen können.

$$\text{A1: } i > i_{\max} \quad (5.44)$$

$$\text{A2: } \left| U^{(i)} - U^{(i-1)} \right| < \delta \quad (5.45)$$

Die maximale Iterationszahl  $i_{\max}$  begrenzt die Anzahl der Durchläufe und damit wesentlich den Rechenaufwand. Wird diese Zahl zu gering gewählt, besteht die Gefahr, dass die aktuelle Lösung weit vom Optimum entfernt ist. In der Abbruchbedingung A2 wird geprüft, ob die Stellgrößenänderung zwischen zwei Iterationen einem Mindestmaß  $\delta$  entspricht. Ist dies nicht der Fall, ändern sich die Stellgrößen nur noch marginal und die Berechnung kann abgebrochen werden.

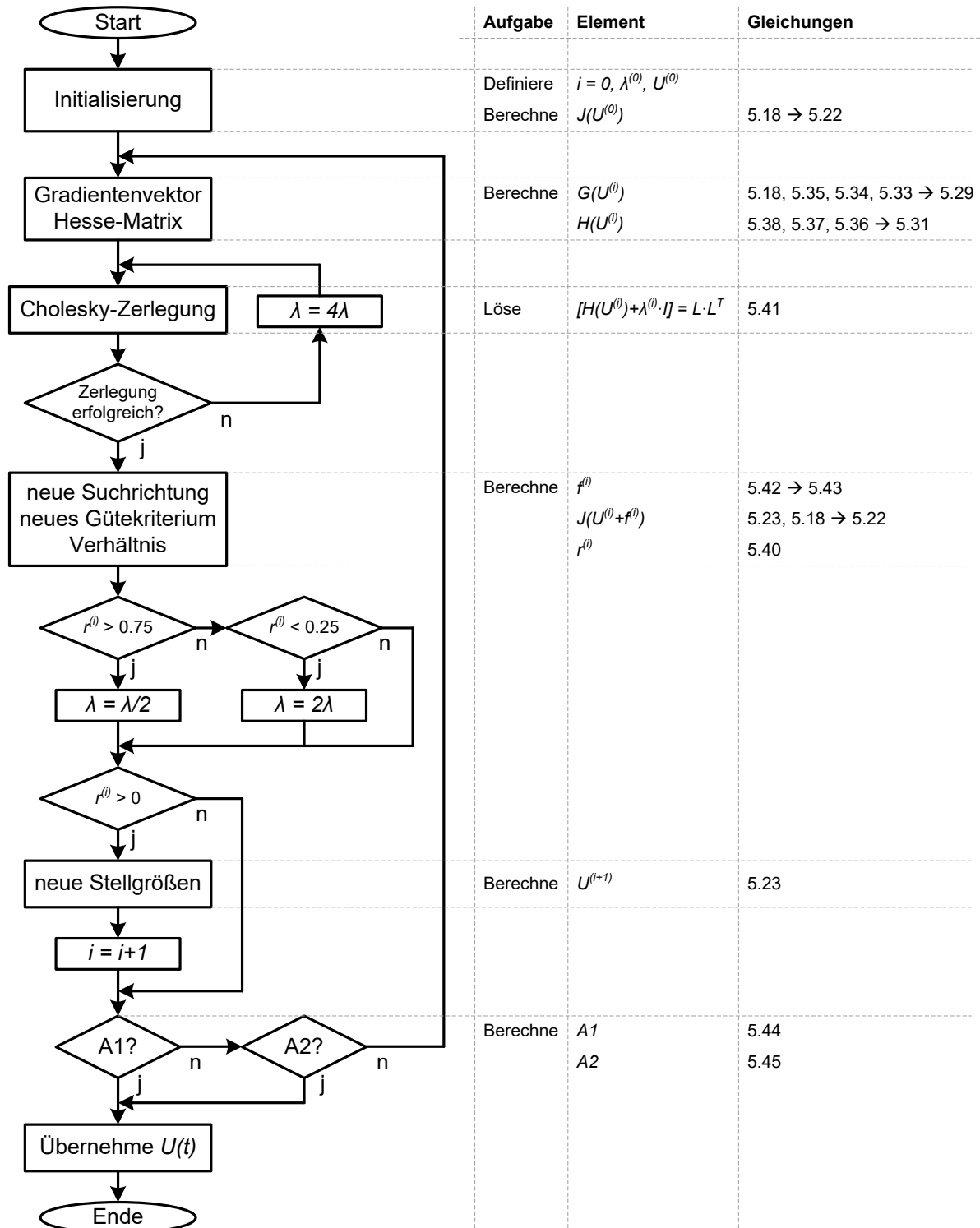


Abbildung 5.10: Ablaufplan und Berechnungsanleitung zum Newton-basierten Levenberg-Marquardt-Algorithmus zur Berechnung optimaler Stellgrößen

### 5.2.5 Wahl der Reglerparameter

Für den beschriebenen Optimierungsalgorithmus stehen mehrere Reglerparameter zur Verfügung. Im Gegensatz zu konventionellen Reglern, bei denen zumeist Proportional-, Integral- und Differentialanteil zur Einstellung eines gewünschten Regelkreisverhaltens festgelegt werden, sind es hier Parameter, die vornehmlich den Optimierungsalgorithmus begrenzen. Vor allem die Auswahl der Zeithorizonte hat entscheidenden Einfluss auf das Regelkreisverhalten. Zum einen dürfen die Zeithorizonte nicht so groß gewählt werden, dass die benötigte Rechenzeit zur Lösung der Optimierungsaufgabe den verfügbaren Zeitrahmen überschreitet und zum anderen müssen die Zeithorizonte groß genug sein, um dem trägen Zeitverhalten des Glasprozesses hinreichend gerecht zu werden. Für die Wahl der Reglerparameter finden sich in der Fachliteratur unterschiedliche Empfehlungen, die nachfolgend auf den vorliegenden Anwendungsfall übertragen werden.

#### **$N_1$ - Beginn vom Prädiktionszeitraum**

In [70] wird empfohlen, den Prädiktionsbeginn gleich der Zeit zu wählen, mit der die Eingangsgrößen verzögert auf den Prozess wirken. Diese Modell-Totzeit entspricht nicht der Strecken-Totzeit des Prozesses. Bei den hier betrachteten KNN liegt dieser Wert ohnehin bei 1. Ein späterer Beginn des Prädiktionszeitraums macht wenig Sinn. Als Beispiel sei der Prädiktionsbeginn  $N_1 = 5$ . Zur Vorhersage der Regelgröße für den Zeitraum  $(t + 6)$  benötigt das KNN mit  $TDL = 10$  die zurückliegenden Werte der Stell- und Regelgrößen. Für die Zeitpunkte  $(t + 1)$  bis  $(t + 5)$  würden diese Werte zum aktuellen Zeitpunkt  $t$  allerdings nicht vorliegen. Aus diesem Grund wird  $N_1 = 1$  gewählt.

#### **$N_2$ - Ende vom Prädiktionszeitraum**

Die Wahl dieses Parameters gestaltet sich schwierig. In der Fachliteratur finden sich verschiedene Faustformeln. In [70] wird empfohlen  $N_2$  entweder größer dem TDL-Wert oder gleich der Anstiegszeit des Prozesses zu setzen. In [73] gilt die Summe aus der Anstiegszeit und dem Stellhorizont als geeignet. Eine Erhöhung von  $N_2$  bringt eine Erhöhung des Rechenaufwands mit sich, jedoch nicht in dem Maße wie eine Erhöhung des Stellhorizonts. Da alle Stellgrößen nach dem Stellhorizont als konstant angenommen werden, können die prädizierten Regelgrößen und deren partielle Ableitungen im Prädiktionshorizont vergleichsweise schnell berechnet werden. Ein längerer Prädiktionszeitraum ist aus diesem Grund möglich und empfehlenswert.

Die neuronalen Netze 1 bis 4 sind mit  $TDL = 10$  konfiguriert, was einem zurückliegenden Zeithorizont von fünf Minuten entspricht. Für die Prädiktion wird ein deutlich größerer Zeitraum angestrebt. Um die Trägheit des Glasherstellungsprozesses und damit die Wirkung der zu optimierenden Stellgrößen annähernd zu erfassen, wird ein Prädiktionshorizont von 30 Minuten ( $N_2 = 60$ ) empfohlen. Dieser Wert ist auch mit den Ergebnissen zur Multi Step Prediction konform (vgl. Abschnitt 4.3).

### **$N_u$ - Ende vom Stellhorizont**

Die Wahl basiert auf einem Kompromiss. Mit der Anzahl zu optimierender Stellgrößen steigt die Dimension des Optimierungsproblems und damit der Rechenaufwand zur Lösung. Ist der Prädiktionszeitraum groß, scheint die Optimierung nur weniger Stellgrößen kaum aussagekräftig. Auf der anderen Seite muss die Frage beantwortet werden, ob die Berechnung einer längeren Stellgrößenfolge Sinn macht, wenn ohnehin nur das erste Element dieser Stellfolge an den Prozess ausgegeben wird. Zudem wird die Wahl durch den verfügbaren zeitlichen Rahmen begrenzt, in dem der gesamte Optimierungsalgorithmus gelöst werden muss. Die Faustformeln und Empfehlungen gehen in der Fachliteratur weit auseinander, z. B.  $N_u = 1$  [77],  $5 \leq N_u \leq 20$  [73] oder  $N_u = n$  (entspricht hier TDL) [70]. Auch hier sollte die Wahl des Parameters der Trägheit des Systems und zugleich der Netzkonfiguration gerecht werden. Wird nur eine Stellgröße optimiert, sind alle anderen zeitlich folgenden Stellgrößen bis zum Ende des Prädiktionszeitraums konstant und damit die Vorhersagen des KNN wenig aussagekräftig. Konstante Stellgrößen sind beim vorliegenden Prozess aus physikalischer Sicht ohnehin nicht realistisch. Aus diesem Grund wird als Stellhorizont ein Bereich zwischen  $\frac{1}{3} N_2 \leq N_u \leq \frac{1}{2} N_2$  angestrebt, z. B.  $N_u = 20$  (10 min).

### **$\rho$ - Wichtungsfaktor Stellaufwand**

Ein Wert von  $\rho = 1$  bedeutet, dass die Priorität der Minimierung der Gütefunktion gleichermaßen auf die Summe der quadratischen Regelabweichungen und die Summe der Stellgrößenänderungen verteilt ist. In *MATLAB* liegt der Wert in den aufgeführten Beispielen bei 0,05 [67], in [70] wird aus Gründen der numerischen Robustheit lediglich ein Wert größer Null empfohlen. Eine Erhöhung dieses Parameters hat zur Folge, dass die Stellgrößenänderungen stärker "bestraft" werden [73]. Dies führt zu geringen Änderungen der Stellgrößen und damit zu einem trägeren Regelkreisverhalten. Die in den evaluierten KNN verwendeten Stellgrößen, insbesondere die Gemeindefuhr BATCH und die Gesamtgasmenge GFC, unterliegen teilweise sprungförmigen Änderungen. Der Wertebereich der prozentualen Geschwindigkeit der Gemeindefuhr liegt im gesamten Messzeitraum zwischen 35 - 40 % (vgl. Anhang D). Damit auch zukünftige Stellgrößen in diesem Bereich



liegen, können zusätzliche Nebenbedingungen formuliert werden, die die Stellgrößen beschränken. Um das Signalverhalten der jeweiligen Stellgrößen beizubehalten, sollte demnach ein kleiner Wert für  $\rho$  gewählt werden, z. B.  $\rho = 0,1$ .

### Nebenbedingungen

Ein Erfolgsfaktor von MPC-Regelungen ist die Implementierung von Nebenbedingungen. Diese Einschränkungen für Stell- und Regelgrößen erlauben eine prozessspezifische Anlagenfahrweise und erhöhen die Sicherheit der Regelung. Besonders bei verfahrenstechnischen Prozessen, bei denen Regel- und Stellgrößen (z. B. Volumenströme) physikalischen und geometrischen Begrenzungen unterliegen, hat sich diese Methodik, in der Fachliteratur auch als "constrained MPC" bezeichnet, bewährt. Die zeitabhängigen Nebenbedingungen für Stell- und Regelgrößen lassen sich als (einseitig offene oder geschlossene) Ungleichungen für vorgesehene Bereiche formulieren. In der Praxis kommt den Ungleichungen für Stellgrößen eine größere Bedeutung zu. Hier sind es vor allem sicherheitskritische Aspekte, die berücksichtigt werden, wohingegen Verletzungen der Nebenbedingungen für Regelgrößen mitunter hingenommen werden können.

$$\text{Für Regelgrößen: } u_{\min}(t) \leq u(t+k) \leq u_{\max}(t) \quad \text{für } k = 0, 1, \dots, N_u - 1. \quad (5.46)$$

$$\text{Für Stellgrößen: } y_{\min}(t) \leq y(t+k) \leq y_{\max}(t) \quad \text{für } k = 1, \dots, N_2. \quad (5.47)$$

Für den Glasherstellungsprozess lassen sich verschiedene Nebenbedingungen formulieren. Zur Wahrung der hohen Qualitätsanforderungen können neben den Sollwertverläufen für die qualitätsbestimmenden Prozessgrößen auch zusätzliche Grenzbereiche definiert werden, die nicht verlassen werden dürfen, insbesondere bei Sollwertänderungen. Eine allgemeingültige Formulierung für WCT2 wäre z. B.

$$\text{WCT2: } [r_{WCT2}(t) - 2K] \leq \hat{y}_{WCT2}(t+k) \leq [r_{WCT2}(t) + 2K]. \quad (5.48)$$

Ein Sollwert von 1.230 K würde bedeuten, dass die zukünftigen Stellgrößen so berechnet werden müssten, dass die Regelgröße im Prädiktionszeitraum zwischen 1.228 - 1.232 K liegt. Die gleiche Vorgehensweise ist für die Prozessgröße GLASS denkbar.

Bei den Stellgrößen können Erfahrungswerte und physikalische Grenzen berücksichtigt werden. Wie bereits erwähnt, liegt der Wert der Stellgröße BATCH im gesamten Messzeitraum zwischen 35 - 40 %. Ein sinnvoller Ansatz wäre es demzufolge, die zukünftigen Stellgrößen ebenfalls auf diesen Bereich zu begrenzen, zumal aus dem vorliegenden

Datensatz nicht erkennbar ist, ob eine langsamere oder schnellere Mengezufuhr überhaupt realisierbar ist und welche Auswirkungen dadurch hervorgerufen werden.

$$\text{BATCH:} \quad 35\% \leq u_{\text{BATCH}}(t+k) \leq 40\% \quad (5.49)$$

Die Implementierung von Nebenbedingungen erfolgt direkt im Gütekriterium. Ein bekannter Ansatz zur Lösung dieses restringierten Optimierungsproblems ist das Lagrange-Verfahren, welches sich auf den vorgestellten Optimierungsansatz anwenden lässt. Zur Lösung von Optimierungsproblemen mit Nebenbedingungen sei an dieser Stelle auf die einschlägige Fachliteratur verwiesen, insbesondere auf [82].

### Abbruchbedingungen

Im vorgestellten Optimierungsalgorithmus sind zwei Abbruchbedingungen integriert, bei deren Erfüllung die Suche nach der optimalen Stellgrößenfolge beendet wird. Eine Erweiterung ist ohne großen Aufwand möglich. Mit der maximalen Iterationszahl wird die Anzahl an Durchläufen begrenzt. Besonders bei flachen Minima kann es zu sehr vielen Iterationen kommen, in denen jeweils der Gradientenvektor und die Hesse-Matrix berechnet werden müssen. Das Training der KNN wurde mit dem Levenberg-Marquardt-Algorithmus nach durchschnittlich neun Iterationen beendet (vgl. Abschnitt 3.2.2). Mit dem Quasi-Newton-Algorithmus waren 54 Iterationen notwendig, beim klassischen Gradientenverfahren lag der Wert sogar bei 1.000, was der maximal zulässigen Iterationsanzahl entsprach. Der LM-Algorithmus benötigte im Schnitt 0,23 Sekunden zur Lösung der Trainings-Optimierungsaufgabe. Die Anzahl der zu optimierenden Stellgrößen bei der Regelung ist deutlich geringer als die Anzahl der zu optimierenden Gewichte beim Training, weshalb die maximale Iterationszahl ohne Bedenken auf beispielsweise  $i_{\max} = 20$  reduziert werden kann. Um sicherzustellen, dass der Rhythmus der Regelung im laufenden Betrieb nicht gefährdet wird, könnte eine weitere mögliche Abbruchbedingung die tatsächliche Zeit berücksichtigen. Alle 30 Sekunden werden neue Messwerte eingelesen und Stellgrößen ausgegeben. Zur Sicherstellung des Regelungsbetriebs wäre es daher sinnvoll, nach spätestens 25 Sekunden die Optimierung abubrechen und die bis dato ermittelte Stellgrößenfolge zu übernehmen. Durch die Verwendung des LM-Algorithmus wird eine Abstiegsrichtung erzwungen, weshalb die Stellgrößenfolge in diesem Fall nicht optimal ist, aber dennoch nahe dem Optimum liegt.

## 5.3 Diskussion

In den vorangegangenen Abschnitten wurde ein Konzept zur neuronalen prädiktiven Regelung von Prozessgrößen in Flachglaswannen abgeleitet. Ein Praxistest des abgeleiteten Regelkonzeptes an einer realen Flachglasanlage konnte nicht realisiert werden. Die Auswahl des Gütekriteriums, die Auswahl der Optimierungsmethode und die Wahl der Reglerparameter basiert auf den Erkenntnissen aus der Identifikation der Künstlichen Neuronalen Netze, persönlichen Gesprächen mit Experten aus der Glasindustrie und den Recherchen in themengebundener Fachliteratur, u. a. [40, 64, 67, 70, 71, 73]. Das Thema der Optimierung ist sehr umfangreich und bietet vielfältige Herangehensweisen. Das vorgestellte Regelkonzept kann als eine von mehreren geeigneten Lösungen verstanden werden. Insbesondere die Wahl der Reglerparameter und der Optimierungsmethode bieten Raum für Diskussionen. Nachfolgend werden Schwierigkeiten, Grenzen und Alternativen zur Identifikation der KNN und zur abgeleiteten Regelstrategie beschrieben.

- **Nachteil LM-Algorithmus**

Zur Berechnung der neuen Suchrichtung muss die Matrix  $[H(U^{(i)}) + \lambda^{(i)}I]$  positiv definit sein. Das bedeutet, dass in jeder Iteration erst die aufwendig zu berechnende Hesse-Matrix  $H(U^{(i)})$  aufgestellt und anschließend mit der Cholesky-Zerlegung die gesamte Matrix auf positive Definitheit überprüft werden muss. Bei ungünstiger Konditionierung kann es sein, dass diese Schritte innerhalb einer Iteration mehrmals durchgeführt werden müssen (vgl. Abbildung 5.10). Dadurch kann die benötigte Rechenzeit schnell ansteigen, was zu Problemen bei der Echtzeitregelung führen kann. Für die Implementierung dieser rechenintensiven Optimierungsmethode müssen somit ausreichende Rechenkapazitäten auf den Steuereinheiten bereit gestellt werden, über dessen Umfang an dieser Stelle keine abschließende Aussage getroffen werden kann.

- **Quasi-Newton-Algorithmus als Alternative**

Beim Quasi-Newton-Algorithmus wird die Hesse-Matrix approximiert, da deren exakte Kenntnis nicht zwingend notwendig ist. Ein bekannter Algorithmus zur Approximation der Hesse-Matrix ist der von Broyden-Fletcher-Goldfarb-Shanno. Durch die Approximation verringert sich der Rechenaufwand zur Berechnung der Hesse-Matrix erheblich. Allerdings wird zur effektiven Optimierung ein zusätzlicher Algorithmus zur Bestimmung der Schrittweite (z. B. mit einem Gradientenverfahren) benötigt. In [70]

wird dieser Ansatz ausführlich beschrieben und empfohlen. Beim Training lieferte der Quasi-Newton-Ansatz (Funktion *trainbfg*, vgl. Abschnitt 3.2.2) nur eine durchschnittliche Performance.

- **Abtastzeit**

Die Abtastzeit des Datensatzes und damit des aufgestellten Regelkonzeptes beträgt 30 Sekunden. Alternativ könnte die Abtastzeit erhöht werden, z. B. auf eine Minute. Dadurch vergrößert sich der zeitliche Rahmen zur Lösung der Optimierungsaufgabe und deutlich längere Prädiktionszeiträume wären möglich.

- **Wahl der Reglerparameter**

Die Auswahl der Reglerparameter basiert auf in der Fachliteratur gefundenen allgemeingültigen Empfehlungen und bereitgestellten Informationen zum Glasherstellungsprozess. Auf Erfahrungen zu Reglereinstellungen für MPC-Regelungen auf Basis von KNN in der Glasindustrie konnte nicht zurückgegriffen werden. Die gewählten Parameter können als erster Ansatz verstanden werden. Eine simulative Überprüfung ist zwingend zu empfehlen.

- **Fortwährendes Training**

Im Gegensatz zu konventionellen Regelkreisen wird bei MPC-Regelungen das hinterlegte Prozessmodell auch in der Arbeitsphase genutzt. Umso wichtiger ist es, dass dieses Modell hinreichend genau ist. Um die aktuelle Prozessdynamik stets zu berücksichtigen, sollten die KNN in regelmäßigen Abständen (z. B. alle vier Stunden) neu trainiert werden. Auf diese Weise werden sich ändernde Informationen zu Stell-, Regel- und nicht messbaren Störgrößen direkt im Prozessmodell berücksichtigt. Die durchschnittliche Trainingszeit der evaluierten Netze liegt bei 0,23 Sekunden. Damit sind auch noch kürzere Intervalle denkbar.

- **Messfehler und Identifikation bei laufendem Betrieb**

Der verwendete Datensatz wurde unterschiedlich aufbereitet, allerdings standen keine Informationen zu verwendeten Messmethoden und -geräten zur Verfügung. Der Datensatz konnte nicht von eventuellen Messfehlern und Ungenauigkeiten bereinigt werden. Ein weiterer kritischer Diskussionsansatz ist die Identifikation der KNN aus Daten des laufenden Betriebs. Üblicherweise erfolgt die Identifikation des dynamischen Verhaltens nach bestimmten Testeingangssignalen (z. B. Sprungfunktionen der Eingangsgrößen). Diese Anlagentests sind bei diesem Prozess zu aufwendig und kaum möglich. Aus diesem Grund wurde auf die Identifikation des Prozessverhaltens im laufenden Betrieb zurückgegriffen.

- **Feuerwechsel**

Alle 15 Minuten findet ein Feuerwechsel statt. Dieser Wechsel, der bis zu 60 Sekunden dauern kann, muss berücksichtigt und übergeordnet gesteuert werden. Denkbar wäre ein Aussetzen der Optimierungsschleife unmittelbar vor dem Feuerwechsel, da die vom Regelalgorithmus berechnete Stellgröße ohnehin nicht ausgeführt werden kann.

- **Konvergenz und Stabilität**

Das Regelkonzept und die Reglerparameter konnten nicht an einer realen Flachglasanlage getestet werden. Aus diesem Grund sind bisher keine belastbaren Aussagen zur Konvergenzgeschwindigkeit und Stabilität des geschlossenen Regelkreises möglich. Das vorgestellte Prinzip der "Regelung durch Optimierung" führt bei einem hinreichend genauen Prozessmodell für gewöhnlich zur Stabilität. Die Implementierung von Nebenbedingungen gewährleistet zusätzlich das Einhalten von gestellten Sicherheitsanforderungen.

- **Erweiterung auf Mehrgrößensysteme**

Der Optimierungsansatz wurde auf die evaluierten KNN übertragen, die allesamt nur eine Ausgangsgröße besitzen. Eine Erweiterung für den MIMO-Fall ist ohne großen Aufwand möglich. Unterschiede bestehen nur in der Dimensionierung der Vektoren und Matrizen. Eine mathematische Beschreibung einer prädiktiven Regelung für Mehrgrößensysteme findet sich beispielsweise in [73].

In diesem Kapitel wurde ein geeignetes regelungstechnisches Konzept vollständig beschrieben und hinsichtlich Grenzen und Alternativen diskutiert. Für den vorliegenden Anwendungsfall stehen somit erstmalig prozessbezogene Richtlinien zur Verfügung, die die Umsetzung einer prädiktiven Regelstrategie mit neuronalen Prozessmodellen erleichtern.

---

## 6 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurden Künstliche Neuronale Netze zur Berechnung von Prozessgrößen in Flachglasschmelzwannen modelliert und ein Konzept einer neuronalen prädiktiven Regelstrategie vorgestellt. Die Identifikation der KNN erfolgte auf Basis eines Datensatzes einer realen Flachglasanlage.

Die industrielle Herstellung von Flachglas ist ein anspruchsvoller und komplexer Prozess, geprägt durch Totzeiten, große Zeitverzögerungen und sensible thermochemische Vorgänge. Trotz des hohen Grades der Basisautomatisierung greifen auch heute noch erfahrene Wannenführer manuell in den Glasherstellungsprozess ein, um Unregelmäßigkeiten auszugleichen. Die bisher verwendeten Prozessmodelle können das komplexe Prozessverhalten nur begrenzt wiedergeben oder sind für Regelungszwecke ungeeignet. Die in dieser Arbeit evaluierten Künstlichen Neuronalen Netze ermöglichen dem Anlagenfahrer nun Prozessmodelle zu verwenden, die auf gelerntes Prozesswissen zurückgreifen, denn KNN sind allein aus Beobachtung des Prozesses in der Lage, dessen Verhalten anzunehmen. Damit stehen erstmalig Prozessmodelle zur Verfügung, die mit einem nachgewiesenen hohen Genauigkeitsbereich eine für den Glasherstellungsprozess unabdingbare vorrausschauende Anlagenfahrweise ermöglichen.

Der Schwerpunkt dieser Arbeit lag im Entwurf und in der Parametrierung geeigneter Netzstrukturen, um die qualitätsbestimmenden Prozessgrößen in einer Flachglasschmelzwanne hinreichend genau berechnen zu können. Der Einfluss der Netzparameter, u. a. der Trainingsmethode, der Aktivierungsfunktion, der Neuronenanzahl und der Anzahl zurückliegender Messwerte, wurde eingehend untersucht. Zur systematischen Evaluation wurde eine grafische Benutzeroberfläche programmiert, die eine übersichtliche Konfiguration aller relevanten Netzparameter ermöglicht und die Simulationsergebnisse grafisch darstellt. Dieser Konfigurationsassistent kann zur Realisierung der vorgestellten prädiktiven Regelstrategie als zusätzliches Werkzeug auf eine reale Anlage übertragen werden, um eventuelle Änderungen der Netzstrukturen vorzunehmen und das Training der Netze zu veranlassen.

Ein Ergebnis dieser Arbeit ist, dass in *MATLAB* standardmäßig definierte oder von vergleichbaren interdisziplinären Anwendungen übernommene Parameterwerte nicht auf diesen anspruchsvollen Prozess übertragbar sind. Die Simulationen haben gezeigt, dass auf

Begrenzungseigenschaften von Aktivierungsfunktionen verzichtet werden kann. Mit einer klassischen linearen Aktivierungsfunktion lassen sich signifikant bessere Ergebnisse erzielen. Die Tauglichkeit des Levenberg-Marquardt-Algorithmus zum Training der neuronalen Netze konnte für diesen Prozess nachgewiesen werden. Gleichwohl bot eine gezielte Modifikation der Trainingsfunktion auf konstante Initialisierung eine vollständige Reproduzierbarkeit der Trainingsergebnisse. Damit konnte die Tauglichkeit der neuronalen Netze für eine prädiktive Regelstrategie, bei der in regelmäßigen Abständen ein erneutes Training beabsichtigt ist, deutlich erhöht werden.

Ein weiteres Ergebnis ist die Abhängigkeit zum Rechenaufwand. Netze, die eine rechenintensive Struktur aufweisen, z. B. durch eine hohe Anzahl an Neuronen oder Tapped Delay Lines, erzeugen verrauschte und unbrauchbare Ausgangssignale. Es hat sich weiterhin gezeigt, dass Netze mit nur einer Ausgangsgröße eine höhere Genauigkeit erzielen, als Netze, die gleichzeitig mehrere Ausgangsgrößen berechnen. Ein weiterer Vorteil gegenüber einem umfassenden MIMO-Modell besteht darin, dass sich die kleineren MISO-Netze je nach Bedarf gezielt in bestehende Regelstrukturen einfacher implementieren lassen.

Durch die systematische Herangehensweise konnten Netz-Konfigurationen ermittelt werden, mit denen die relevanten Prozessgrößen mit einer sehr hohen Genauigkeit berechnet werden können. Insbesondere die Testergebnisse zur One Step Prediction sind durchweg zufriedenstellend. Die Standardabweichung vom NARX-Netz 1, welches die Decktemperatur WCT2 am Ende der Abstehwanne berechnet, beträgt im zwölfstündigen Testzeitraum nur 0,0534 K, was bei Temperaturen von durchschnittlich 1.229 K sehr beachtlich ist. Der maximale Fehler im gesamten Testzeitraum beträgt bei optimaler Konfiguration nur 0,2163 K. Der Glasstand am Ende der Abstehwanne konnte mit einem NIO-Netz (Netz 4) ebenfalls sehr genau berechnet werden. Die Standardabweichung liegt bei 0,0607 mm. Damit stehen dem Anwender Konfigurationshinweise und Prozessmodelle zur Verfügung, die für die Berechnung ausgewählter Größen prädestiniert sind.

Im zweiten Teil der Arbeit wurde ein regelungstechnisches Konzept abgeleitet. Eine Analyse bestehender Regelstrukturen mit Künstlichen Neuronalen Netzen und ein Abgleich mit den Prozessbedingungen in einer Flachglasschmelzwanne führten zu einer prädiktiven Regelstrategie. Die mathematische Beschreibung zur Berechnung der Vorhersagewerte für die Regelgrößen wurde detailliert hergeleitet. Die Berechnung der zukünftigen Stellgrößen erfolgt nach dem Newton-basierten Levenberg-Marquardt-Algorithmus. Bereits beim Training der KNN erzielte dieser Ansatz die mit Abstand besten Ergebnisse. Desweiteren wurde eine prozessbezogene Vorauswahl der einzustellenden Reglerparameter getroffen. Somit

existieren erstmalig Richtlinien zum Entwurf einer neuronalen prädiktiven Regelstrategie zur Optimierung des Glasherstellungsprozesses.

Eine abschließende Bewertung des aufgestellten Regelkonzeptes könnte durch einen Praxistest an einer realen Glaswanne erfolgen. In weiterführenden Arbeiten sollten vorab erste Erkenntnisse simulativ gewonnen werden. Eine Möglichkeit wäre es, ein zweites trainiertes Künstliches Neuronales Netz als "reale" Regelstrecke zu verwenden. Auf diese Weise kann der Einfluss der einzelnen Reglerparameter untersucht werden. Ein weiterer zukünftiger Ansatz, um die Genauigkeit der Netze zu erhöhen, könnte in einer zusätzlichen Gewichtung der zurückliegenden Messwerte liegen. Erst kürzlich gemessene oder ausgegebene Regel- und Stellgrößen sollten höher bewertet werden als weiter zurückliegende Messwerte. Eine weitere Möglichkeit zur Prozessoptimierung besteht in der Kombination aus Künstlichen Neuronalen Netzen und CFD-Modellen als Softsensoren. Die KNN könnten somit auf qualitätsbestimmende Prozessgrößen trainiert werden, deren Messung in der Praxis nicht möglich oder fehlerbehaftet ist.

Aufgrund des Mehrgrößencharakters, der komplexen Dynamik des Prozesses und des bereits hohen Grades der Basisautomatisierung lässt sich abschließend sagen, dass die in dieser Arbeit evaluierten Netze und vorgestellten Ansätze die bestehenden und konventionellen regelungstechnischen Verfahren sinnvoll ergänzen können.

*"Essentially all models are wrong, but some models are useful"*

- George Edward Pelham Box -



---

## Anhang

### A Trainingsmethoden in *MATLAB*

In *MATLAB* stehen zwölf verschiedene Algorithmen zum Training von KNN zur Verfügung.

Tabelle A.1: Verfügbare Trainingsfunktionen in *MATLAB*

Funktion	Name (englisch)
trainlm	Levenberg-Marquardt
trainbr	Bayesian Regularization
trainbfg	BFGS Quasi Newton
trainrp	Resilient Backpropagation
trainscg	Scaled Conjugate Gradient
traincgb	Conjugate Gradient with Powell/Beale Restarts
traincgf	Fletcher-Powell Conjugate Gradient
traincgp	Polak-Ribière Conjugate Gradient
trainoss	One Step Secant
traingdx	Variable Learning Rate Gradient Descent
traingdm	Gradient Descent with Momentum
traingd	Gradient Descent

Nach [67] ist die Levenberg-Marquardt-Funktion die schnellste Optimierungsfunktion und besonders zur Funktionsanpassung und Mustererkennung geeignet. Die Quasi-Newton-Methode (*trainbfg*) ist annähernd schnell. Für große Netze (Anzahl Wichtungsfaktoren > 1.000) verschlechtert sich die Performance beider Methoden, da sie mehr Speicherkapazität und eine längere Rechenzeit benötigen. Für das Training von großen Netzen sind zudem die Algorithmen Scaled Conjugate Gradient (*trainscg*) und Resilient Backpropagation (*trainrp*) geeignet. Sie benötigen weniger Speicher und sind dennoch schneller als die Standard-Gradientenverfahren.

Levenberg-Marquardt-Algorithmus zum Training der KNN

Pro Schrittweite wird die zu minimierende MSE-Funktion durch eine quadratische Funktion ersetzt und das Minimum berechnet. Bei den konventionellen quadratischen Trainingsalgorithmen ist es üblich, die aufwendig zu berechnende Hesse-Matrix  $H$  aufzustellen. In *MATLAB* wird beim Training der Netze mit dem LM-Algorithmus eine Approximation dieser Matrix genutzt. Die Hesse-Matrix und der Gradientenvektor  $G$  berechnen sich nach

$$H \approx J^T \cdot J, \quad (\text{A.1})$$

$$G = J^T \cdot E. \quad (\text{A.2})$$

Die Jacobi-Matrix  $J$  enthält die partiellen Ableitungen der Fehler  $e$  nach den Gewichten  $w$ . Die Fehler sind im Vektor  $E$  zusammengefasst. Die Matrix ergibt sich zu

$$J = \begin{bmatrix} \frac{\partial e(u_1, w)}{\partial w_1} & \dots & \frac{\partial e(u_1, w)}{\partial w_W} \\ \vdots & \ddots & \vdots \\ \frac{\partial e(u_N, w)}{\partial w_1} & \dots & \frac{\partial e(u_N, w)}{\partial w_W} \end{bmatrix}. \quad (\text{A.3})$$

Darin sind  $u$  die Eingangsgrößen,  $N$  die Anzahl aller gewichteten Eingangswerte (inkl. TDL etc.) und  $W$  die Gesamtanzahl der Gewichte und Schwellwerte. Die Gleichung vom LM-Algorithmus lautet dann

$$(J^T \cdot J + \lambda \cdot I) \cdot w = J^T \cdot E. \quad (\text{A.4})$$

Die Gewichtsänderung berechnet sich in *MATLAB* nach

$$w_{i+1} = w_i - (J^T \cdot J + \lambda \cdot I)^{-1} \cdot J^T \cdot E. \quad (\text{A.5})$$

Für  $\lambda = 0$  entspricht die Gleichung dem bekannten Newton-Verfahren. Wird dagegen  $\lambda$  groß gewählt, entspricht die Gleichung dem Gradientenverfahren mit einer kleinen Schrittweite. In der Nähe des Fehlerminimums ist das Newton-Verfahren schneller und genauer als das Gradientenverfahren [67]. Demzufolge wird  $\lambda$  nach jedem Iterationsschritt schnellstmöglich reduziert. Der LM-Parameter wird nur erhöht, wenn ein vorangegangener Iterationsschritt den quadratischen Fehler vergrößert hat. Dann wirkt das Gradientenverfahren zur Minimierung der MSE-Funktion.

## Anhang

### B Aktivierungsfunktionen in *MATLAB*

Tabelle B.1 listet alle verfügbaren Aktivierungsfunktionen sowie deren Kennlinie, mathematische Beschreibung und mögliche Verwendungszwecke auf.

Tabelle B.1: Verfügbare Aktivierungsfunktionen in *MATLAB*

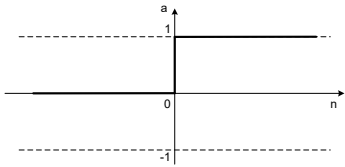
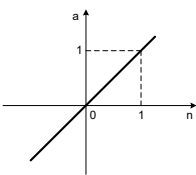
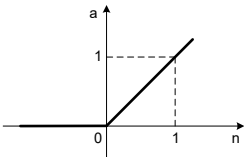
Funktionen mit Zweipunktverhalten	hardlim	Hard limit transfer function	
			$n \geq 0 \rightarrow a = 1$ $n < 0 \rightarrow a = 0$
	hardlims	<ul style="list-style-type: none"> <li>zur Klassifikation und Mustererkennung</li> <li>sprunghaft, nichtlinear, unstetig, nicht differenzierbar</li> </ul>	
		Symmetric hard limit transfer function	
Lineare Funktionen	purelin		$a = n$
		<ul style="list-style-type: none"> <li>zur Funktionsapproximation</li> <li>linear, stetig, differenzierbar</li> </ul>	
	poslin	Positive linear transfer function	
			$n \geq 0 \rightarrow a = n$ $n < 0 \rightarrow a = 0$
		<ul style="list-style-type: none"> <li>zur Funktionsapproximation</li> <li>negativ begrenzt, stückweise linear, stetig, begrenzt differenzierbar</li> </ul>	

Tabelle B.1: Verfügbare Aktivierungsfunktionen in *MATLAB* (Fortsetzung 1)

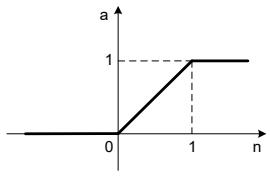
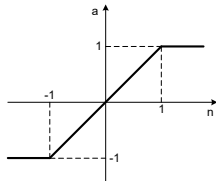
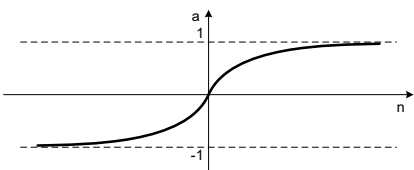
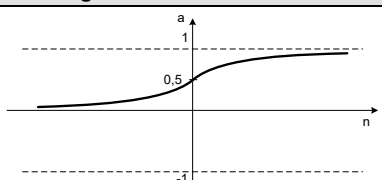
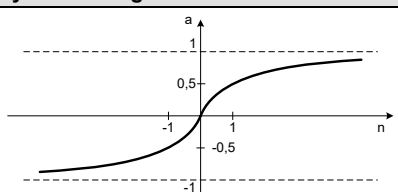
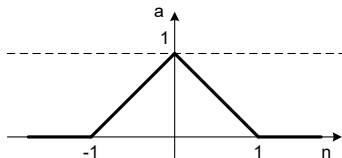
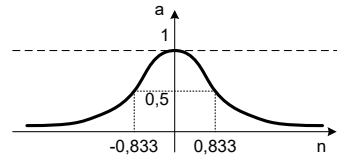
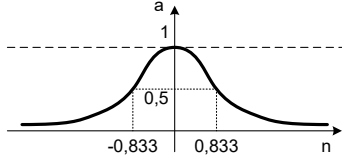
Lineare Funktionen	satlin	<b>Saturating linear transfer function</b>	
			$n < 0 \rightarrow a = 0$ $0 \leq n \leq 1 \rightarrow a = n$ $n \geq 1 \rightarrow a = 1$
	satlins	<b>Symmetric saturating linear transfer function</b>	
			$n < -1 \rightarrow a = -1$ $-1 \leq n \leq 1 \rightarrow a = n$ $n \geq 1 \rightarrow a = 1$
Sigmoide Funktionen	tansig	<b>Hyperbolic tangend sigmoid transfer function</b>	
			$a = \frac{2}{1 + e^{-2n}} - 1 = \frac{1 - e^{-2n}}{1 + e^{-2n}}$ $(= \tanh(n))$
	logsig	<b>Logarithmic sigmoid transfer function</b>	
			$a = \frac{1}{1 + e^{-n}}$
	elliotsig	<b>Elliot symmetric sigmoid transfer function</b>	
			$a = \frac{n}{1 +  n }$
		<ul style="list-style-type: none"> <li>zur Time Series Prediction</li> <li>schneller als <i>tansig</i>, negativ und positiv begrenzt [-1 1], nichtlinear, stetig, differenzierbar</li> </ul>	

Tabelle B.1: Verfügbare Aktivierungsfunktionen in *MATLAB* (Fortsetzung 2)

Basisfunktionen	tribas	Triangular basis transfer function	
			$n < -1 \rightarrow a = 0$ $-1 \leq n \leq 1 \rightarrow a = 1 -  n $ $n \geq 1 \rightarrow a = 0$
		<ul style="list-style-type: none"> <li>zur Time Series Prediction, Funktionsapproximation</li> <li>Konzentration auf lokalen Eingangsbereich, stückweise linear, stetig, begrenzt differenzierbar</li> </ul>	
	radbas	Radial basis transfer function	
			$a = e^{-n^2}$
		<ul style="list-style-type: none"> <li>zur Time Series Prediction, Funktionsapproximation</li> <li>Konzentration auf lokalen Bereich an Eingangswerten, nichtlinear, stetig, differenzierbar</li> </ul>	
	radbasn	Normalized radial basis transfer function	
			$a_i = \frac{e^{-n_i^2}}{\sum_{k=1}^m e^{-n_k^2}}$ $m = \text{Anzahl Eingangsgrößen}$
		<ul style="list-style-type: none"> <li>zur Time Series Prediction, Funktionsapproximation</li> <li>äquivalent zu <i>radbas</i>, die Ausgangswerte werden allerdings normiert (Division durch die Summe aller Ausgänge)</li> <li>Konzentration auf lokalen Bereich an Eingangswerten, nichtlinear, stetig, differenzierbar</li> </ul>	
Andere	compet	Competitive transfer function	
		$n: \begin{array}{ c c c c } \hline 2 & 1 & 4 & 3 \\ \hline \end{array} \rightarrow a: \begin{array}{ c c c c } \hline 0 & 0 & 1 & 0 \\ \hline \end{array}$	$n_i = \max(n) \rightarrow a_i = 1$ $n_i < \max(n) \rightarrow a_i = 0$
		<ul style="list-style-type: none"> <li>zur Klassifikation und Gruppierung</li> <li>dem Maximalwert der Eingangsgrößen wird der Ausgangswert 1 zugeordnet, dem Rest die 0</li> </ul>	
	softmax	Soft max transfer function	
		$n: \begin{array}{ c c c } \hline 0 & 1 & -0,5 \\ \hline \end{array} \rightarrow a: \begin{array}{ c c c c } \hline 0,167 & 0,495 & 0,101 & 0,275 \\ \hline \end{array}$	$a_i = \frac{e^{n_i}}{\sum_{k=1}^m e^{n_k}}$ $m = \text{Anzahl Eingangsgrößen}$
		<ul style="list-style-type: none"> <li>der exponentielle Wert jeder Eingangsgröße wird durch die Summe der exponentiellen Werte aller Eingangsgrößen dividiert</li> <li>zur Klassifikation und Gruppierung</li> </ul>	
	netinv	Inverse transfer function	
		$n: \begin{array}{ c c c c } \hline 2 & 1 & 4 & 3 \\ \hline \end{array} \rightarrow a: \begin{array}{ c c c c } \hline 0,5 & 1 & 0,25 & 0,33 \\ \hline \end{array}$	$a_i = \frac{1}{n_i}$
		<ul style="list-style-type: none"> <li>zur Klassifikation und Gruppierung</li> <li>jeder Ausgangswert entspricht dem Kehrwert des Eingangswertes</li> </ul>	

## Anhang

### C Aufbereitung der verfügbaren Messgrößen

Die Arbeit basiert auf einem Datensatz einer realen amerikanischen Flachglasanlage. Die Prozessgrößen wurden von den gegebenen physikalischen Einheiten in das in Europa bekannte SI-Einheitensystem überführt. Tabelle C.1 listet für jede physikalische Größe die Gleichungen und Einheiten der Umrechnung auf. In der Tabelle C.2 sind alle verfügbaren Prozessgrößen mit ihren statistischen Kennzahlen aufgelistet.

Tabelle C.1: Umrechnungen in das SI-Einheitensystem

Phys. Größe	gegebene Einheit	Bedeutung	äquivalente SI-Einheit	Bedeutung
Temperatur	°C	Grad Celsius	K	Kelvin
	→ $K = ^\circ C + 273,15$			
Volumenstrom	MSCFH	1000 * Standard Cubic Feet per Hour	Nm³/h	Normkubikmeter pro Stunde
	→ $1 \text{ MSCFH} = 1000 \cdot 1 \text{ SCFH}$			
	→ $1 \text{ MSCFH} = 1000 \cdot \frac{1}{60} \text{ SCFM}$			
	→ $1 \text{ MSCFH} = \frac{1000}{60} 10,7272 \text{ Nm}^3 / \text{h}$			
	→ $1 \text{ MSCFH} = 178,7866667 \text{ Nm}^3 / \text{h}$			
Druck	INH2O	Inches of Water	Pa	Pascal
	→ $1 \text{ INH2O} = 2,49089 \text{ mbar}$			
	→ $1 \text{ INH2O} = 2,49089 \cdot 100 \text{ Pa}$			
	→ $1 \text{ INH2O} = 249,089 \text{ Pa}$			
Länge	in	Inch	mm	Millimeter
	→ $1 \text{ in} = 2,54 \text{ cm}$			
	→ $1 \text{ in} = 25,4 \text{ mm}$			

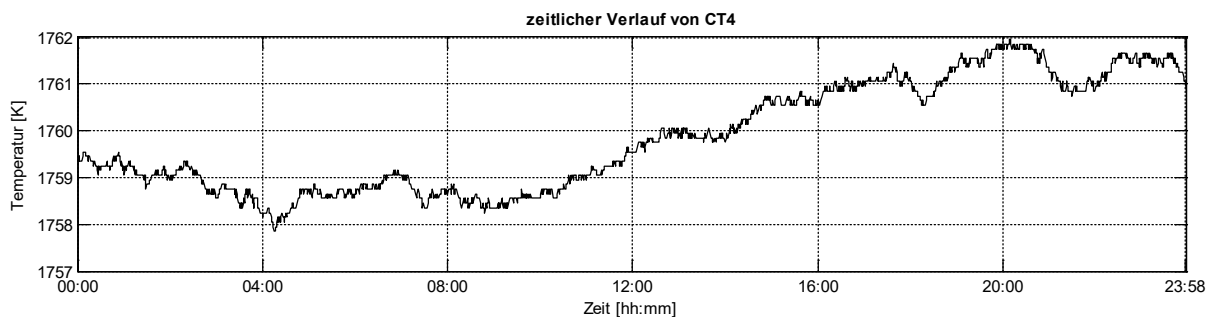
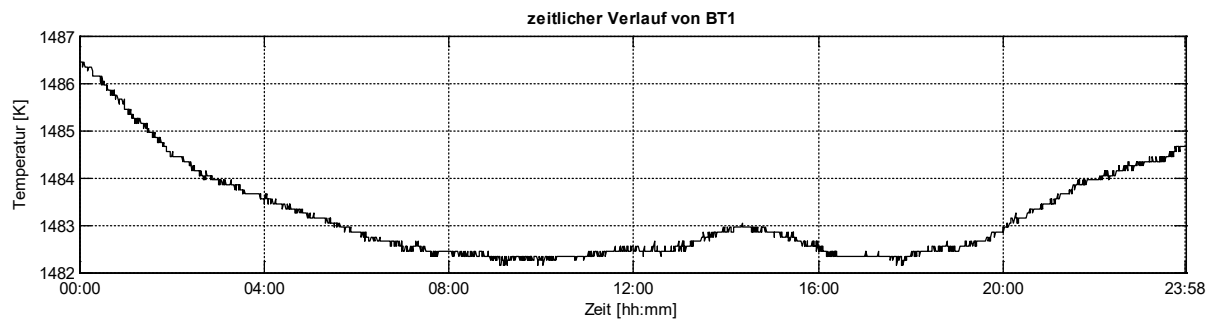
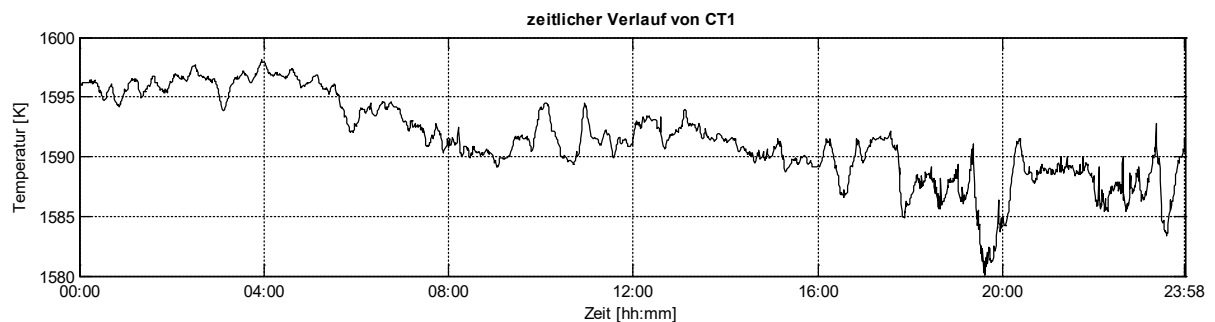
Tabelle C.2: Kennzahlen aller verfügbaren Prozessgrößen im gesamten Messzeitraum

Phys. Größe	Name	Einheit	Minimum	Maximum	Differenz	Mittelwert	Standard-abweichung
Temperatur	CT1	K	1.580,05	1.598,15	18,10	1.591,47	3,502
	BT1	K	1.482,15	1.486,45	4,30	1.483,17	0,971
	CT4	K	1.757,85	1.761,95	4,10	1.759,85	1,160
	BT4	K	1.536,85	1.541,15	4,30	1.538,20	0,973
	WCT1	K	1.424,05	1.427,35	3,30	1.425,36	0,877
	WCT2	K	1.227,45	1.231,75	4,30	1.229,73	1,020
	TRANS	K	1.370,80	1.373,55	2,75	1.371,55	0,783
	DAT	K	310,45	316,35	5,90	312,79	1,594
Volumen-strom	GFC	Nm³/h	4.776,80	23.854,76	19.077,97	22.693,59	3.833,543
	GF1	Nm³/h	726,58	4.983,40	4.256,81	4.716,27	823,410
	GF2	Nm³/h	841,19	5.013,75	4.172,56	4.718,63	810,411
	GF4	Nm³/h	1.027,22	5.075,18	4.047,97	4.764,22	809,630
	GF5	Nm³/h	1.033,51	4.158,65	3.125,14	3.916,20	615,724
	CAF	Nm³/h	227.485,19	281.903,14	54.417,95	252.442,91	5.224,136
	DAF	Nm³/h	469,56	573,97	104,41	544,06	10,289
Druck	WEP	Pa	9,25	19,95	10,70	16,19	1,418
	MPlleft	Pa	-13,88	14,94	28,82	4,69	4,229
	MPright	Pa	-17,24	13,12	30,35	2,66	4,153
Geschwin-digkeit	DAF1	%	0,00	84,01	84,01	10,69	22,474
	DAF2	%	0,00	64,98	64,98	47,07	22,255
	BATCH	%	35,72	39,39	3,66	37,72	0,691
Füllstand	GLASS	mm	-0,30	0,22	0,52	-0,02	0,078

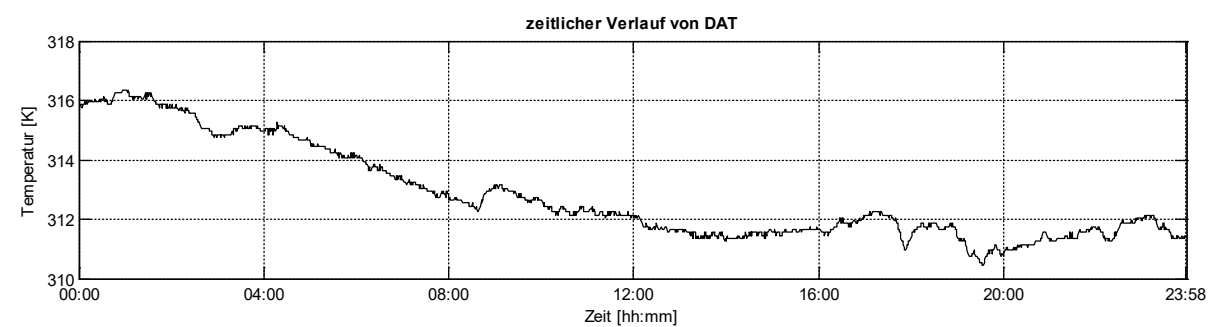
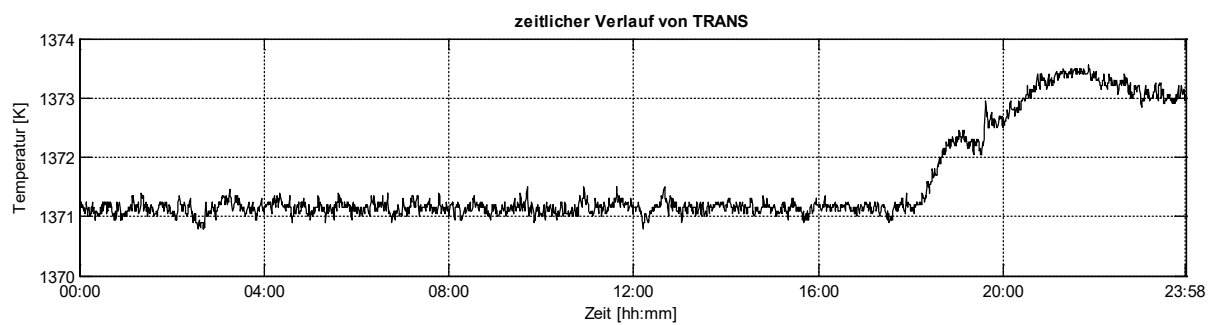
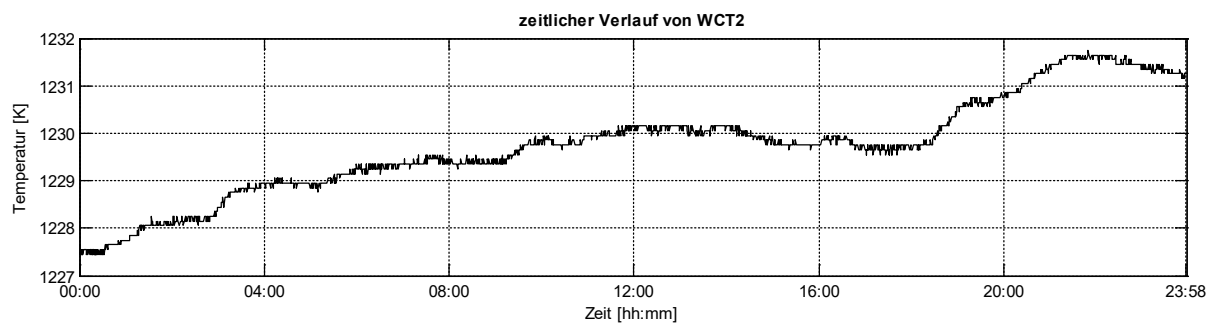
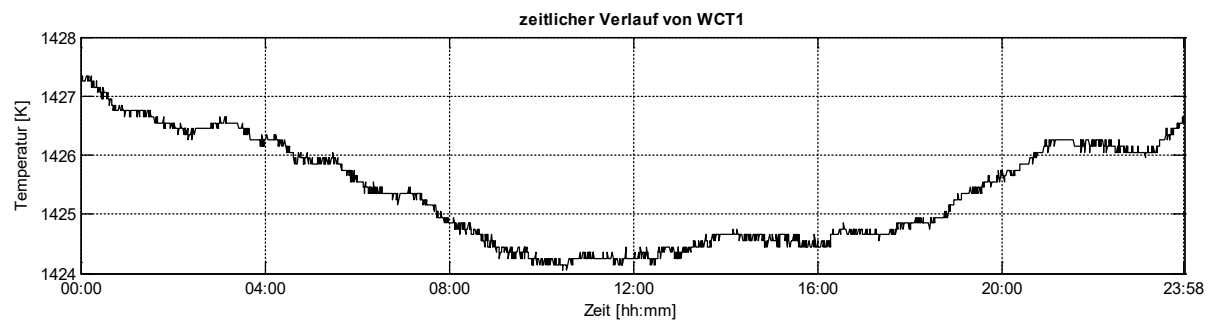
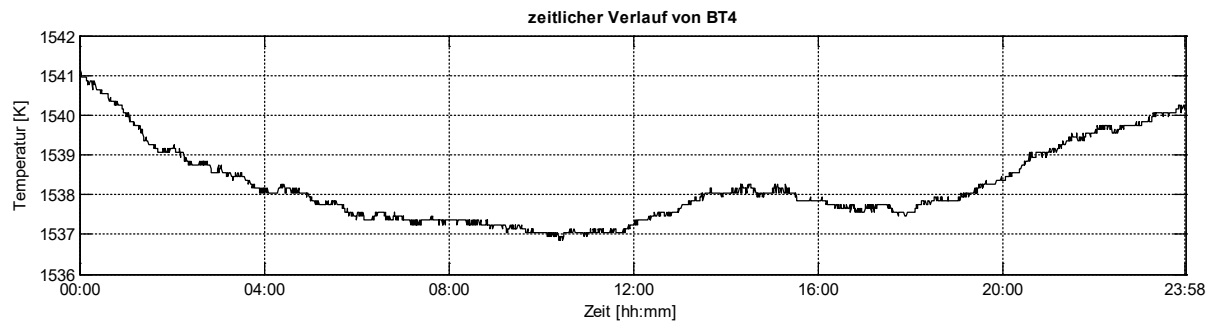
## Anhang

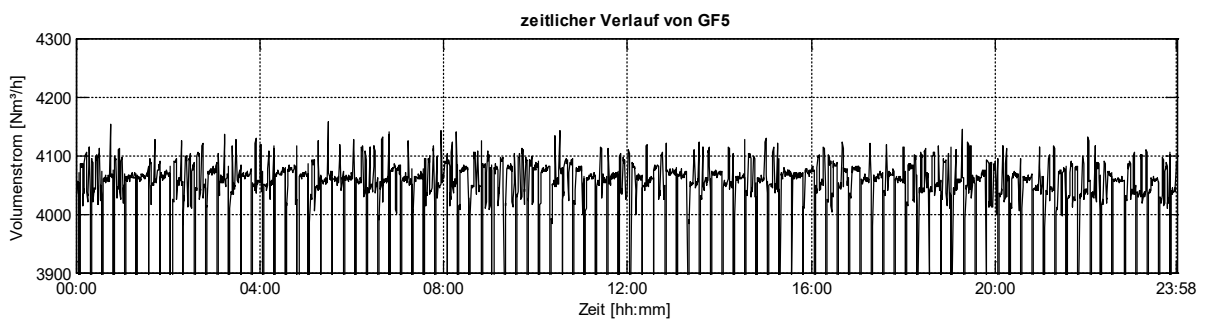
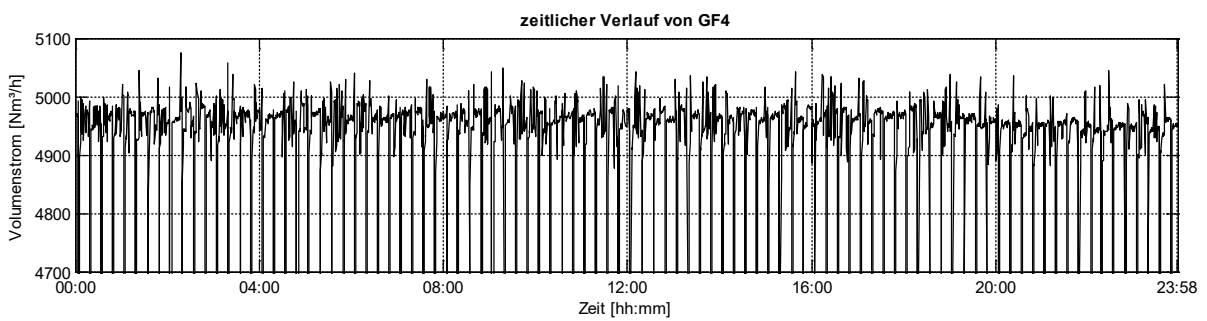
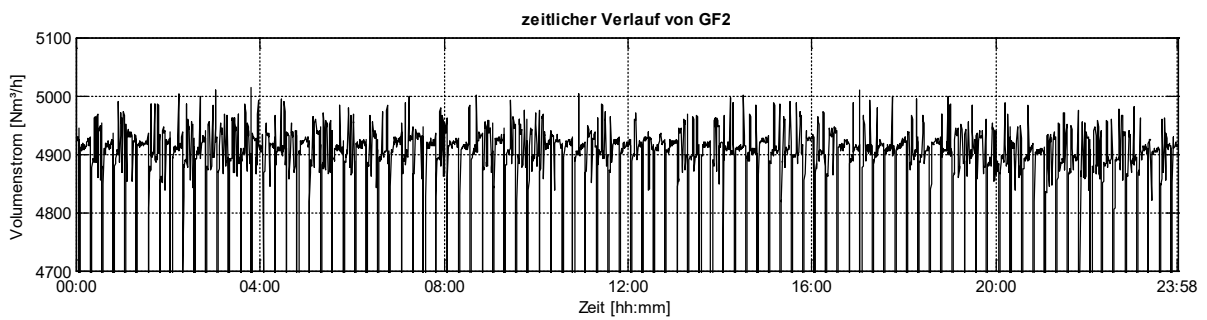
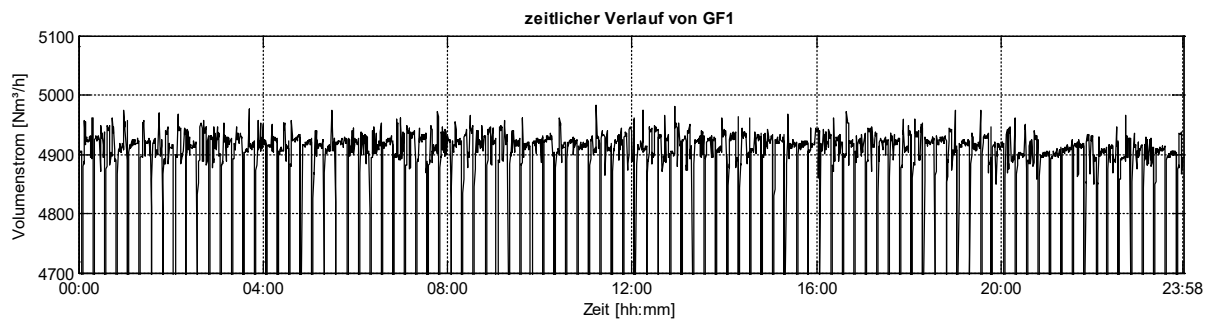
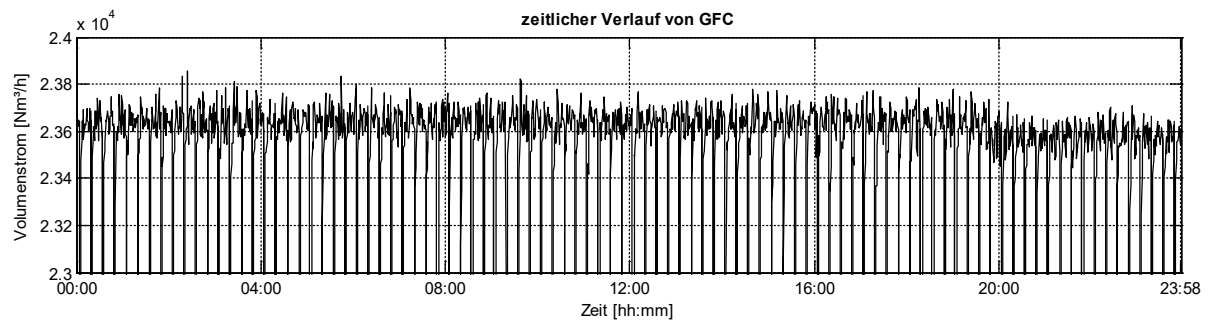
### D Grafische Darstellung aller Messdaten

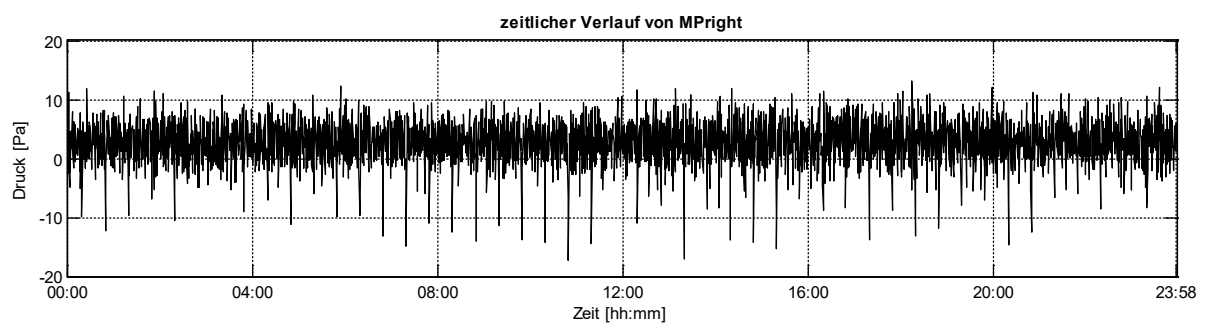
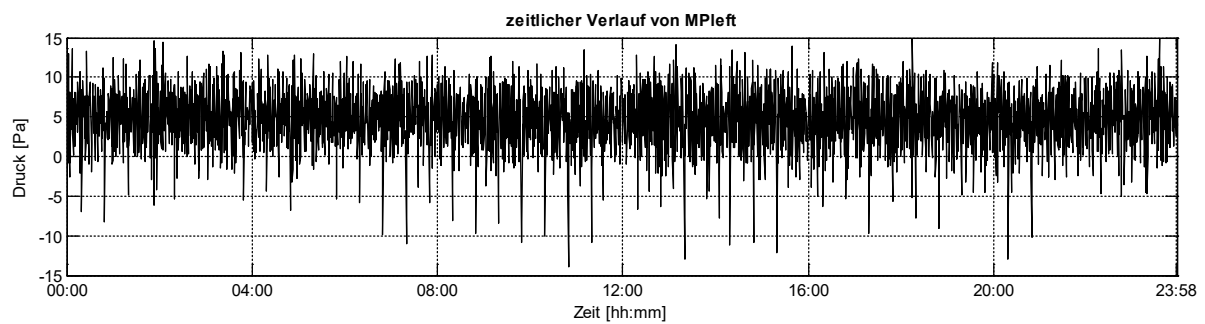
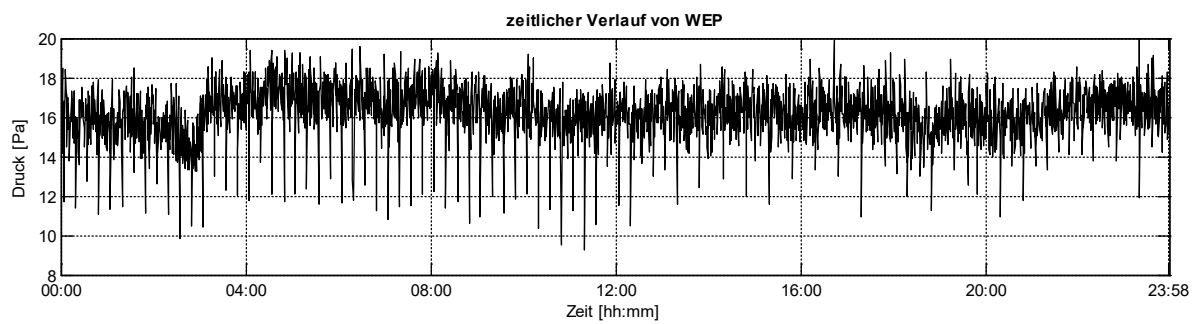
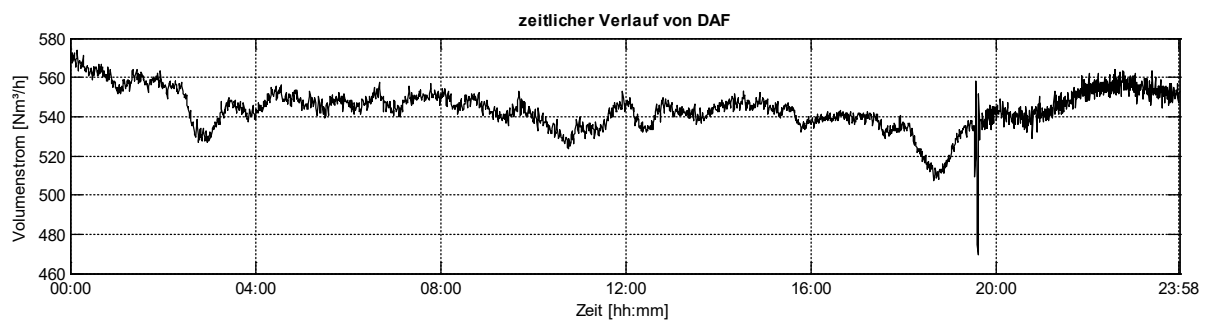
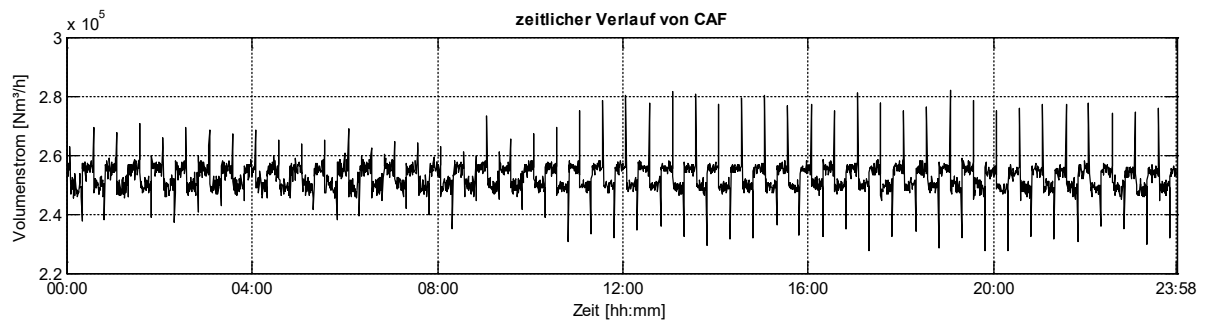
Nachfolgend sind die zeitlichen Verläufe aller verfügbaren Prozessgrößen dargestellt. Die Messdaten sind in das SI-Einheitensystem konvertiert. Es sind ansonsten keine weiteren Änderungen am Datensatz vorgenommen worden. Die Reihenfolge der Abbildungen entspricht der Reihenfolge in Tabelle C.2. Auf Beschriftungen der einzelnen Abbildungen wird an dieser Stelle verzichtet. Die dargestellte Prozessgröße ist im Titel des Diagramms angegeben.

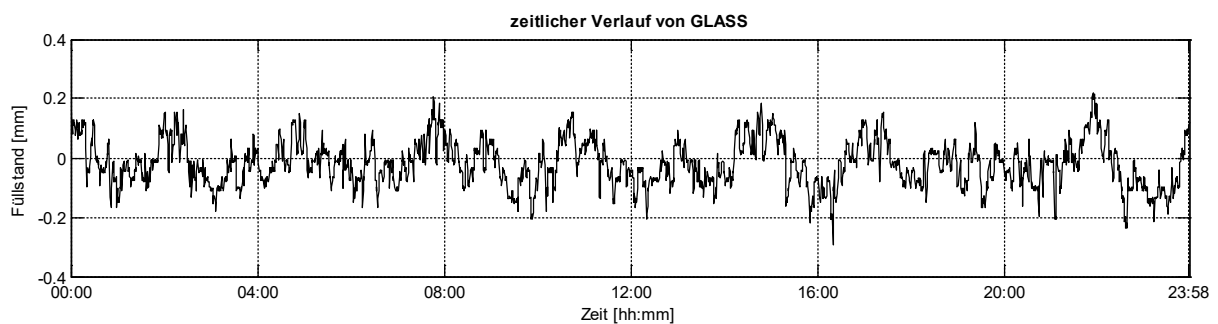
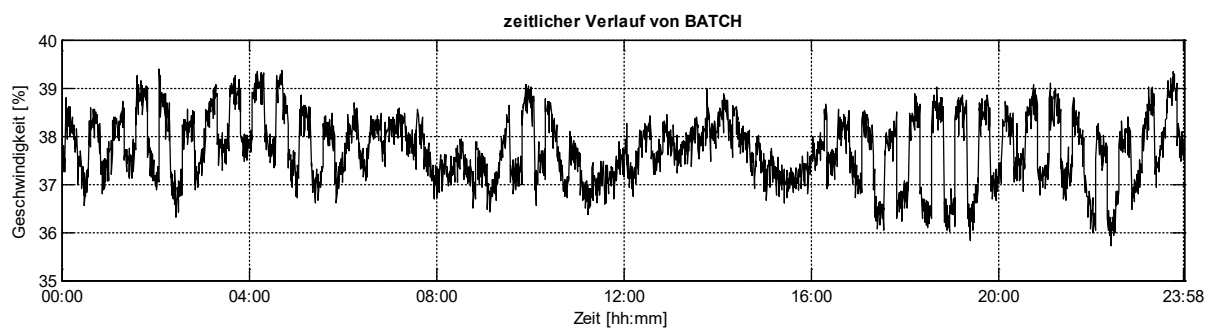
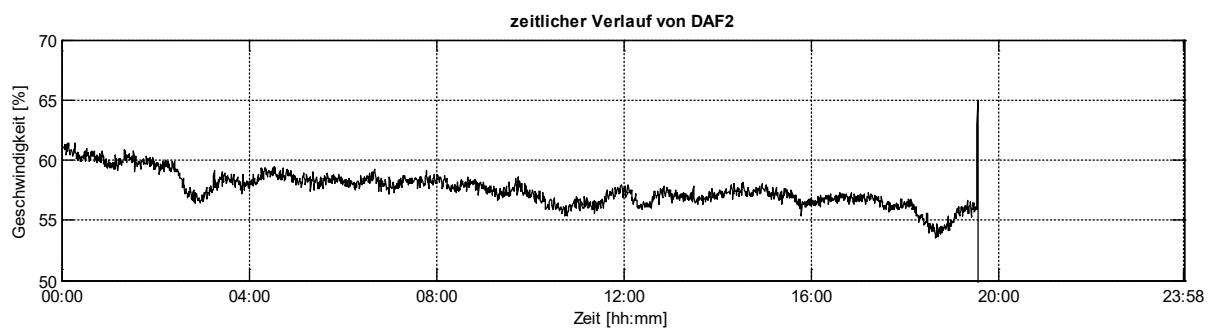
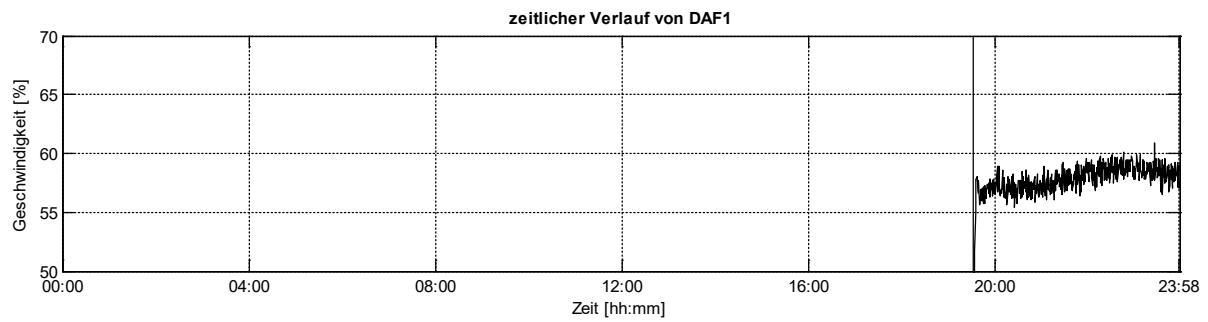












## Anhang

### E Berechnungsbeispiel WCT1

Als Beispiel wird die Temperatur WCT1 (Netz 2) zum Zeitpunkt 18:00 Uhr mit vier verschiedenen Konfigurationen berechnet. Ziel ist es, den Unterschied zwischen der linearen Aktivierungsfunktion *purelin* und der sigmoiden Aktivierungsfunktion *tansig* in den einzelnen Komponenten zu identifizieren. Die Wichtungsfaktoren wurden in *MATLAB* ausgelesen und die Ein- und Ausgangsgrößen auf den Wertebereich des Trainingsdatensets normiert.

#### Berechnungsgrundlage:

Normierte Eingangswerte der Neuronen in Schicht 1:

$$n^1_{norm}(t) = IW^{1,1} \cdot u_{norm}(t) + LW^{1,2} \cdot y_{norm}(t) + b^1 \quad (E.1)$$

Normierte Ausgangswerte der Neuronen in Schicht 1:

$$\text{mit } purelin: \quad a^1_{norm}(t) = n^1_{norm}(t) \quad (E.2)$$

$$\text{mit } tansig: \quad a^1_{norm}(t) = \tanh(n^1_{norm}(t)) \quad (E.3)$$

Normierter Eingangswert vom Neuron in Schicht 2:

$$n^2_{norm}(t) = LW^{2,1} \cdot a^1_{norm}(t) + b^2 \quad (E.4)$$

Normierter Ausgangswert vom Neuron in Schicht 2:

$$a^2_{norm}(t) = n^2_{norm}(t) \quad (E.5)$$

Ausgangswert WCT1 für 18:00 Uhr:

$$y(t) = a^2(t) \quad (E.6)$$

**Beispielrechnung 1:**      **TDL = 2,  $S^1 = 1$ , *purelin* / *purelin***Normierte Eingangswerte der Neuronen in Schicht 1:

$$n^1_{norm}(t) = \begin{pmatrix} 0,0034 \\ 0,0038 \\ -0,0106 \\ 0,0022 \end{pmatrix} \cdot \begin{pmatrix} 0,9796 \\ 0,9724 \\ -0,5227 \\ -0,4608 \end{pmatrix} + \begin{pmatrix} -2,159 & -0,5456 \end{pmatrix} \begin{pmatrix} -0,5152 \\ -0,5758 \end{pmatrix} + (-0,6034) = \underline{(0,8349)}$$

Normierte Ausgangswerte der Neuronen in Schicht 1:

$$a^1_{norm}(t) = \underline{(0,8349)}$$

Normierter Eingangswert vom Neuron in Schicht 2:

$$n^2_{norm}(t) = (-0,3691) \cdot (0,8349) + (-0,2267) = \underline{(-0,5349)}$$

Normierter Ausgangswert vom Neuron in Schicht 2:

$$a^2_{norm}(t) = \underline{(-0,5349039878)}$$

Ausgangswert WCT1 für 18:00 Uhr:

$$\underline{\underline{y(t) = 1424,8174154441K}}$$

**Beispielrechnung 2:** TDL = 2,  $S^1 = 10$ , *purelin* / *purelin*Normierte Eingangswerte der Neuronen in Schicht 1:

$$n^1_{norm}(t) = \begin{pmatrix} -0,0233 & 0,6785 & 0,1976 & 0,4286 \\ -0,7791 & 0,9123 & -0,9728 & 0,3395 \\ ,1663 & 0,4273 & -0,1606 & 0,2392 \\ -0,5114 & -0,2189 & -0,3436 & -0,8255 \\ -0,2387 & 0,1487 & -0,6831 & -0,8857 \\ 0,1732 & -0,7633 & -0,6345 & -0,3388 \\ -0,5274 & 0,7338 & -0,1801 & -0,0297 \\ -0,3823 & 0,8538 & -0,7803 & 0,4177 \\ 0,2370 & 0,6426 & 0,1993 & -0,1764 \\ -0,4645 & -0,4644 & -0,0529 & 0,6549 \end{pmatrix} \cdot \begin{pmatrix} 0,9796 \\ 0,9724 \\ -0,5227 \\ -0,4608 \end{pmatrix} + \begin{pmatrix} 0,4384 & -0,7757 \\ 0,9371 & -0,3073 \\ 0,0625 & -0,0376 \\ -0,3488 & -0,1800 \\ -0,7894 & -0,0333 \\ 0,2209 & 0,7053 \\ 0,5564 & 0,2365 \\ -0,1520 & 0,6496 \\ -0,8182 & 0,2570 \\ -0,4676 & 0,8843 \end{pmatrix} \cdot \begin{pmatrix} -0,5152 \\ -0,5758 \end{pmatrix} + \begin{pmatrix} -0,5114 \\ 0,3410 \\ -0,4290 \\ 0,3638 \\ 0,3858 \\ -0,8601 \\ -0,5077 \\ -0,5314 \\ 0,3373 \\ 0,6915 \end{pmatrix} = \begin{pmatrix} 0,0455 \\ 0,5110 \\ 0,1126 \\ 0,4933 \\ 1,4875 \\ -1,4648 \\ -0,6259 \\ -0,1561 \\ 1,4450 \\ -0,7575 \end{pmatrix}$$

Normierte Ausgangswerte der Neuronen in Schicht 1:

$$a^1_{norm}(t) = \begin{pmatrix} 0,0455 \\ 0,5110 \\ 0,1126 \\ 0,4933 \\ 1,4875 \\ -1,4648 \\ -0,6259 \\ -0,1561 \\ 1,4450 \\ -0,7575 \end{pmatrix}$$

Normierter Eingangswert vom Neuron in Schicht 2:

$$n^2_{norm}(t) = (-0,5806 \quad 0,3725 \quad 0,2567 \quad -0,8230 \quad 0,2873 \quad 0,0994 \quad 0,7960 \quad -0,8438 \quad -0,0692 \quad 0,0500) \cdot \begin{pmatrix} 0,0455 \\ 0,5110 \\ 0,1126 \\ 0,4933 \\ 1,4875 \\ -1,4648 \\ -0,6259 \\ -0,1561 \\ 1,4450 \\ -0,7575 \end{pmatrix} + (-0,0991)$$

$$n^2_{norm}(t) = (-0,5349)$$

Normierter Ausgangswert vom Neuron in Schicht 2:

$$a^2_{norm}(t) = (-0,5349037313)$$

Ausgangswert WCT1 für 18:00 Uhr:

$$y(t) = \underline{\underline{1424,8174158673K}}$$

**Beispielrechnung 3:**      **TDL = 2,  $S^1 = 1$ , *tansig* / *purelin***Normierte Eingangswerte der Neuronen in Schicht 1:

$$n^1_{norm}(t) = \begin{pmatrix} -0,0003 \\ 0,0004 \\ 0,0009 \\ 0,0001 \end{pmatrix} \cdot \begin{pmatrix} 0,9796 \\ 0,9724 \\ -0,5227 \\ -0,4608 \end{pmatrix} + \begin{pmatrix} 0,1932 & 0,0474 \end{pmatrix} \cdot \begin{pmatrix} -0,5152 \\ -0,5758 \end{pmatrix} + (-0,0111) = \underline{\underline{(-0,1383)}}$$

Normierte Ausgangswerte der Neuronen in Schicht 1:

$$a^1_{norm}(t) = \tanh(n^1_{norm}(t)) = \underline{\underline{(-0,1374)}}$$

Normierter Eingangswert vom Neuron in Schicht 2:

$$n^2_{norm}(t) = (4,1948) \cdot (-0,1374) + (0,0410) = \underline{\underline{(-0,5353)}}$$

Normierter Ausgangswert vom Neuron in Schicht 2:

$$a^2_{norm}(t) = \underline{\underline{(-0,5352940534)}}$$

Ausgangswert WCT1 für 18:00 Uhr:

$$\underline{\underline{y(t) = 1424,8167718501K}}$$



**Beispielrechnung 4:** TDL = 2,  $S^I = 10$ , *tansig* / *purelin*Normierte Eingangswerte der Neuronen in Schicht 1:

$$n^1_{norm}(t) = \begin{pmatrix} -0,8106 & 0,5681 & -0,2249 & 0,3173 \\ 0,6065 & 0,0306 & 0,2327 & -0,5259 \\ -0,0126 & -1,5224 & 0,4018 & 1,5397 \\ 0,4243 & -0,2507 & 0,2227 & -0,1534 \\ -0,4315 & 1,2057 & -0,0851 & -0,2755 \\ 1,7905 & 0,7776 & 0,7735 & 0,4841 \\ 0,2520 & 0,1651 & 0,5418 & -0,1275 \\ 0,2566 & -0,0011 & 0,2085 & -0,1457 \\ -0,2057 & 0,2334 & -1,0124 & -1,5029 \\ -0,2933 & 0,5113 & 0,1456 & 0,2991 \end{pmatrix} \cdot \begin{pmatrix} 0,9796 \\ 0,9724 \\ -0,5227 \\ -0,4608 \end{pmatrix} + \begin{pmatrix} -1,2565 & -0,7181 \\ 0,2374 & 0,3013 \\ 0,5420 & -0,6033 \\ 1,3538 & 0,2715 \\ 1,2704 & -0,0098 \\ 0,8262 & -0,7289 \\ -0,1518 & 0,4244 \\ 1,0087 & 0,5768 \\ -0,3507 & 0,5743 \\ 1,2337 & 0,0447 \end{pmatrix} \cdot \begin{pmatrix} -0,5152 \\ -0,5758 \end{pmatrix} + \begin{pmatrix} 2,1877 \\ -0,8845 \\ -1,6378 \\ -0,3881 \\ 0,0842 \\ 0,2387 \\ 0,5081 \\ 1,0479 \\ -0,7779 \\ -1,7941 \end{pmatrix} = \begin{pmatrix} 2,9782 \\ -0,4358 \\ -3,9819 \\ -1,1157 \\ 0,3565 \\ 2,1156 \\ 0,5249 \\ 0,4046 \\ 0,3192 \\ -2,4594 \end{pmatrix}$$

Normierte Ausgangswerte der Neuronen in Schicht 1:

$$a^1_{norm}(t) = \tanh(n^1_{norm}(t)) = \begin{pmatrix} 0,9948 \\ -0,4101 \\ -0,9993 \\ -0,8061 \\ 0,3421 \\ 0,9713 \\ 0,4815 \\ 0,3839 \\ 0,3088 \\ -0,9855 \end{pmatrix}$$

Normierter Eingangswert vom Neuron in Schicht 2:

$$n^2_{norm}(t) = (-0,3188 \quad -0,3201 \quad -0,1426 \quad 0,4419 \quad 0,1402 \quad -0,0264 \quad -0,2811 \quad 0,5314 \quad -0,0874 \quad 0,2324) \cdot \begin{pmatrix} 0,9948 \\ -0,4101 \\ -0,9993 \\ -0,8061 \\ 0,3421 \\ 0,9713 \\ 0,4815 \\ 0,3839 \\ 0,3088 \\ -0,9855 \end{pmatrix} + (-0,0023)$$

$$n^2_{norm}(t) = (-0,5670)$$

Normierter Ausgangswert vom Neuron in Schicht 2:

$$a^2_{norm}(t) = (-0,566980540)$$

Ausgangswert WCT1 für 18:00 Uhr:

$$\underline{\underline{y(t) = 1424,7644903037K}}$$

**Auswertung:**

In der Tabelle E.1 sind verschiedene Kennzahlen einzelner Komponenten zur Berechnung der Ausgangsgröße WCT1 aufgelistet. Ein direkter Vergleich der Kennzahlen für die Schwellwerte  $b^1$ , die Neuroneneingangswerte  $n^1$  und die Wichtungsmatrix  $LW^{2,1}$  verdeutlicht unterschiedliche Wertebereiche. Während bei der Aktivierungsfunktion *purelin* die Komponenten für  $S^1 = 1$  und  $S^1 = 10$  nur geringfügige Abweichungen zeigen (hellgrau schattiert), sind die Unterschiede mit der Tangensfunktion wesentlich größer (dunkelgrau schattiert). Bei den restlichen Komponenten sind die Tendenzen weitestgehend gleich, so dass der Unterschied im Endergebnis auf die beträchtliche Änderung der Schwellwerte  $b^1$  und Wichtungsfaktoren in  $LW^{2,1}$  zurückgeführt werden kann. Der berechnete Ausgangswert für WCT1 unterscheidet sich mit der Aktivierungsfunktion *purelin* lediglich in der sechsten Nachkommastelle. Die Abweichung mit der Aktivierungsfunktion *tansig* beträgt dagegen bereits 0,05 K.

Tabelle E.1: Statistische Kennzahlen der Wichtungsmatrizen und Schwellwerte bei verschiedener Aktivierungsfunktion und Neuronenanzahl zur Berechnung der Ausgangsgröße WCT1

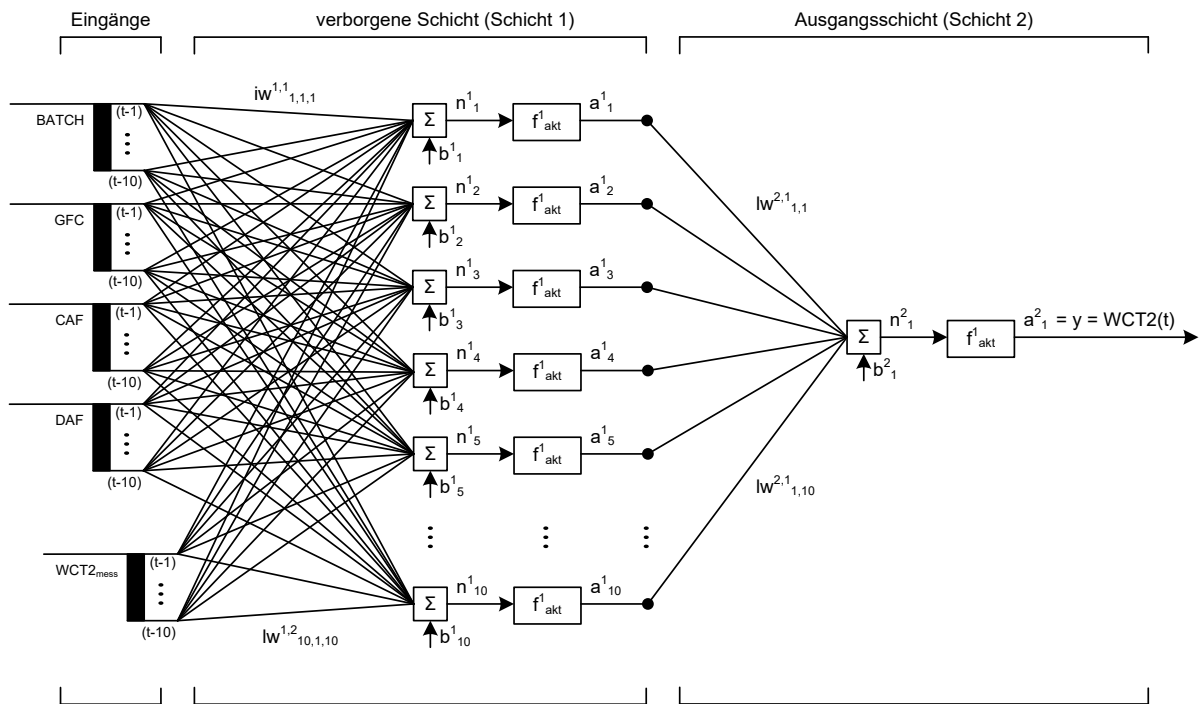
Aktivierungs-funktion	$S^1$	Kennzahl	$IW^{1,1}$	$LW^{1,2}$	$b^1$	$n^1$	$LW^{2,1}$	$b^2$	$n^2$	WCT1 [K]
<i>purelin</i>	1	Summe:	-0,0012	-2,7055	-0,6034	0,8349	-0,3691	-0,2267	-0,5349	1424,817415
		arith. MW	-0,0003	-1,3527	-0,6034	0,8349	-0,3691	-0,2267	-0,5349	
		Gleichrichtwert	0,0050	1,3527	0,6034	0,8349	0,3691	0,2267	0,5349	
	10	Summe:	-2,9868	1,0381	-0,7203	1,0908	-0,4549	-0,0991	-0,5349	1424,817416
		arith. MW	-0,0747	0,0519	-0,0720	0,1091	-0,0455	-0,0991	-0,5349	
		Gleichrichtwert	0,4472	0,4429	0,4959	0,7099	0,4178	0,0991	0,5349	
<i>tansig</i>	1	Summe:	0,0011	0,2406	-0,0111	-0,1383	4,1948	0,0410	-0,5353	1424,816772
		arith. MW	0,0003	0,1203	-0,0111	-0,1383	4,1948	0,0410	-0,5353	
		Gleichrichtwert	0,0004	0,1203	0,0111	0,1383	4,1948	0,0410	0,5353	
	10	Summe:	4,4074	4,8463	-1,4157	-1,2937	0,1695	-0,0023	-0,5670	1424,764490
		arith. MW	0,1102	0,2423	-0,1416	-0,1294	0,0170	-0,0023	-0,5670	
		Gleichrichtwert	0,4892	0,6242	0,9549	1,4691	0,2522	0,0023	0,5670	

## Anhang

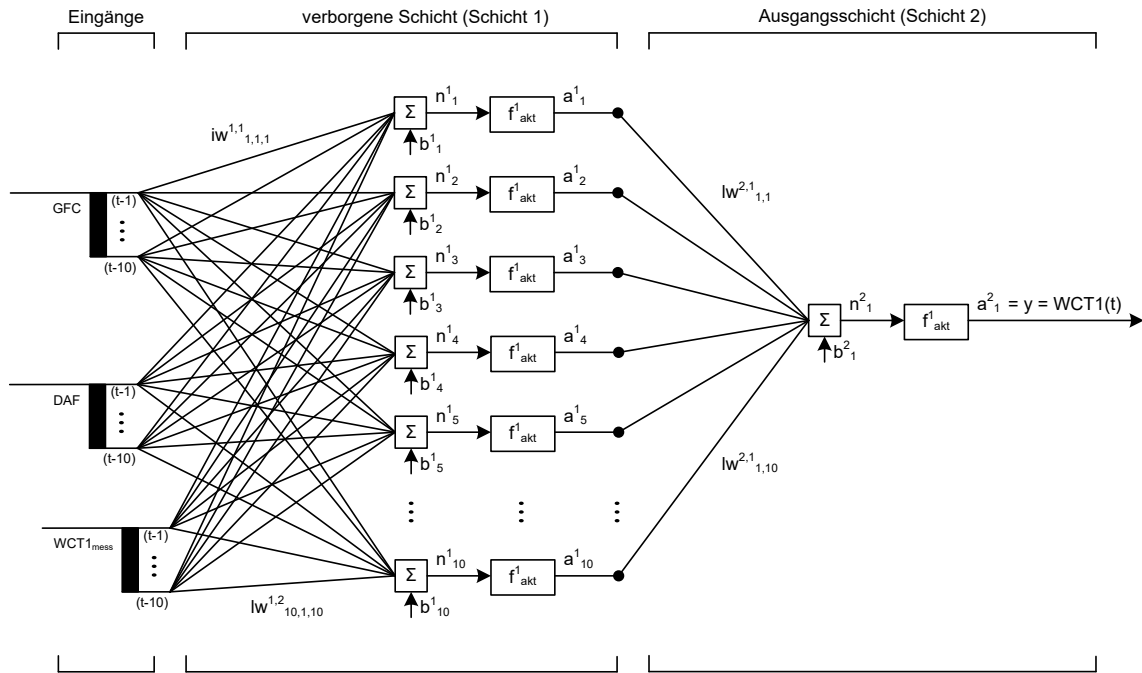
### F Blockschaltbilder und Gleichungen der evaluierten Netze

Die Konfiguration der evaluierten KNN ist in Tabelle 3.8 gegeben. Nachfolgend sind für alle fünf Netze die vollständigen Blockschaltbilder und Gleichungen zur OSP dargestellt. Aus Gründen der Übersichtlichkeit wird auf eine Beschriftung der Abbildungen verzichtet.

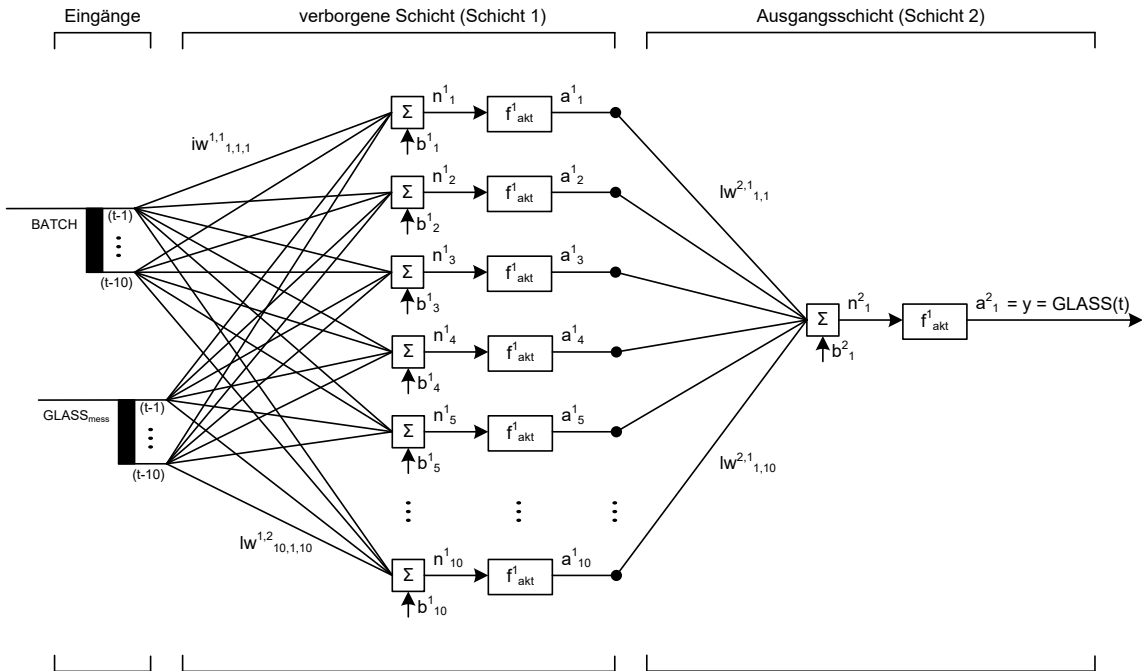
#### Netz 1: NARX, Ausgangsgröße WCT2



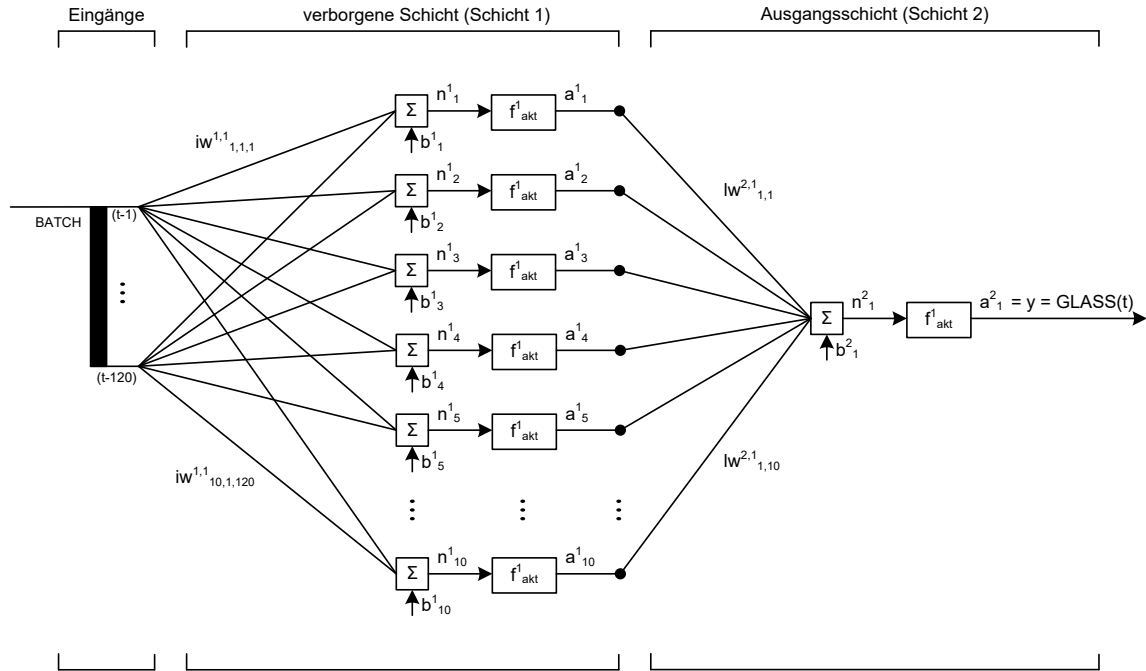
$$WCT2(t) = \left( \sum_{j=1}^{10} lw_{1,j}^{2,1} \left[ \left( \sum_{i=1}^4 \left( \sum_{d=1}^{10} iw_{j,i,d}^{1,1} \cdot u_i(t-d) \right) \right) + \left( \sum_{d=1}^{10} lw_{j,1,d}^{1,2} \cdot WCT2(t-d) \right) + b_j^1 \right] \right) + b_1^2$$

**Netz 2:** NARX, Ausgangsgröße WCT1

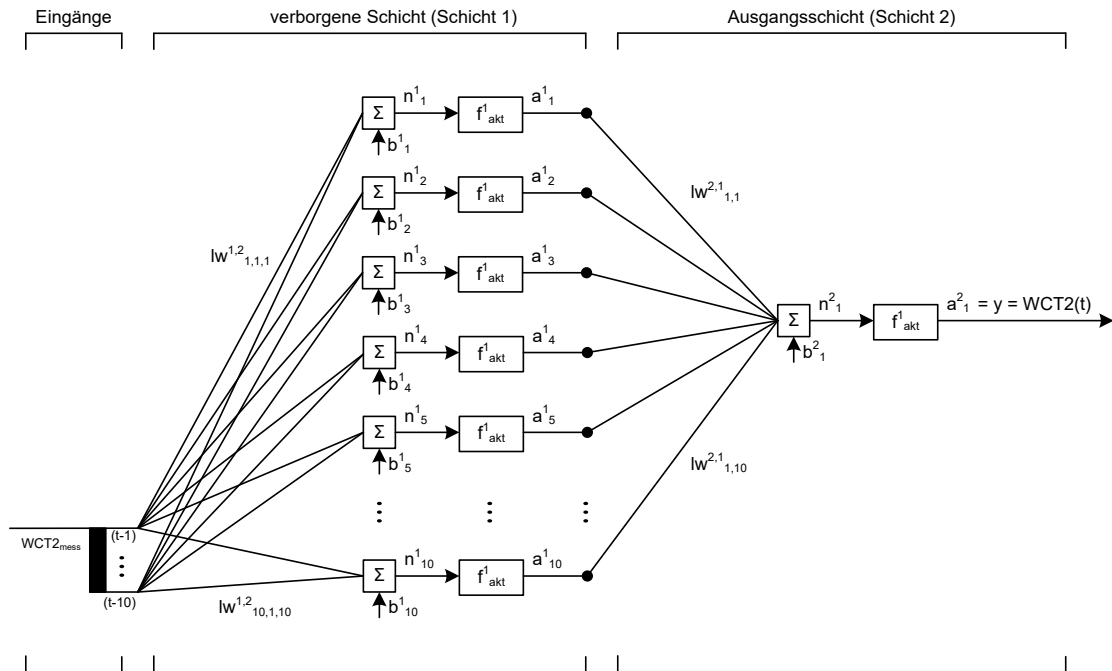
$$WCT1(t) = \left( \sum_{j=1}^{10} lw_{1,j}^{2,1} \left[ \left( \sum_{i=1}^2 \left( \sum_{d=1}^{10} iw_{j,i,d}^{1,1} \cdot u_i(t-d) \right) \right) + \left( \sum_{d=1}^{10} lw_{j,1,d}^{1,2} \cdot WCT1(t-d) \right) + b_j^1 \right] \right) + b_1^2$$

**Netz 3:** NARX, Ausgangsgröße GLASS

$$GLASS(t) = \left( \sum_{j=1}^{10} lw_{1,j}^{2,1} \left[ \left( \sum_{d=1}^{10} iw_{j,i,d}^{1,1} \cdot u_i(t-d) \right) + \left( \sum_{d=1}^{10} lw_{j,1,d}^{1,2} \cdot GLASS(t-d) \right) + b_j^1 \right] \right) + b_1^2$$

**Netz 4:** NIO, Ausgangsgröße GLASS


$$\text{GLASS}(t) = \left( \sum_{j=1}^{10} lw^{2,1}_{1,j} \left[ \sum_{d=1}^{120} iw^{1,1}_{j,i,d} \cdot u_i(t-d) + b^1_j \right] \right) + b^2_1$$

**Netz 5:** NAR, Ausgangsgröße WCT2


$$\text{WCT2}(t) = \left( \sum_{j=1}^{10} lw^{2,1}_{1,j} \left[ \sum_{d=1}^{10} lw^{1,2}_{j,1,d} \cdot \text{WCT2}(t-d) + b^1_j \right] \right) + b^2_1$$

---

## Literatur

- [1] R. Weissmann, "Formgebung von Flachglas", *GLASS: Glasherstellung - Glasveredelung, Prozess und Technologien*, 2. Edition, 2002, S. 108-113.
- [2] H. Müller-Simon, "Industrielle Glasherstellung - Einführung", Vortrag beim Seminar *Industrielle Glasherstellung - Von den Rohstoffen zum Qualitätsprodukt* der Hüttentechnischen Vereinigung der Deutschen Glasindustrie e. V., Offenbach, 12.-13.05.2015.
- [3] P. Boehm, "Herstellungsprozesse für Flachglas", Vortrag beim Seminar *Industrielle Glasherstellung - Von den Rohstoffen zum Qualitätsprodukt* der Hüttentechnischen Vereinigung der Deutschen Glasindustrie e. V., Offenbach, 12.-13.05.2015.
- [4] B. Fleischmann, "Glasschmelzöfen und ihre Konstruktionsmerkmale", Vortrag beim Seminar *Industrielle Glasherstellung - Von den Rohstoffen zum Qualitätsprodukt* der Hüttentechnischen Vereinigung der Deutschen Glasindustrie e. V., Offenbach, 12.-13.05.2015.
- [5] Glassglobal Consulting, "Glass Melting Furnaces", Abbildung einer querbeheizten regenerativen Glasschmelzwanne, zuletzt abgerufen unter <http://www.glassglobal.com/consulting/reports/technology/> am 16.11.2015.
- [6] B. Fleischmann, "Energieeinsatz bei der Glasherstellung", Vortrag beim Seminar *Industrielle Glasherstellung - Von den Rohstoffen zum Qualitätsprodukt* der Hüttentechnischen Vereinigung der Deutschen Glasindustrie e. V., Offenbach, 12.-13.05.2015.
- [7] H. Müller-Simon, "Vorgänge beim Erschmelzen von Glas", Vortrag beim Seminar *Industrielle Glasherstellung - Von den Rohstoffen zum Qualitätsprodukt* der Hüttentechnischen Vereinigung der Deutschen Glasindustrie e. V., Offenbach, 12.-13.05.2015.
- [8] H. Brethfeld et al., *Einführung in die Automatisierungstechnik der Glasindustrie*. Leipzig: VEB Deutscher Verlag für Grundstoffindustrie 1978, S. 158ff.
- [9] F. Hegewald, P. Hemmann und D. Lohann, "Verfahren zur aktuell technologisch geführten Regelung der Oberofenbeheizung von Glasschmelzwannen", Offenlegungsschrift vom VEB Energiekombinat Cottbus, DE3610365A1, Patenterteilung 14.05.1987.

- 
- [10] H. Pieper, "Verfahren zur Regelung der Beheizung von Glas-Wannenöfen", Europäische Patentschrift der Sorg GmbH & Co. KG, EP0748773B1, Lohr am Main, Patenterteilung 21.01.1999.
- [11] T. Hiejima, H. Azuma und M. Sawasaki, "Numerical analysis and measurement of glass flow in a small melting furnace", *Glass Science Technology*, Vol. 76, Nr. 4, 2003, S. 155-165.
- [12] L. Pilon, G. Zhao und R. Viskanta, "Three-dimensional flow and thermal structures in glass melting furnaces: Part 1 - Effects of the heat flux distribution", *Glass Science Technology*, Vol. 75, Nr. 2, 2002, S. 55-67.
- [13] L. Pilon, G. Zhao und R. Viskanta, "Three-dimensional flow and thermal structures in glass melting furnaces: Part 2 - Effects of batch blanket and bubbles", *Glass Science Technology*, Vol. 75, Nr. 3, 2002, S. 115-124.
- [14] O. Auchet, P. Riedinger, C. Iung und O. Malassé, "Simplified First-Principles Modelling of Glass Furnaces for Control Purpose", *International Federation of Automatic Control 2002*, zuletzt abgerufen unter <http://www.nt.ntnu.no/users/skoge/prost/proceedings/dycops04/pdffiles/papers/172.pdf> am 15.03.2016.
- [15] O. Auchet, P. Riedinger, O. Malassé und C. Iung, "First-principles simplified modelling of glass furnaces combustion chambers", *Control Engineering Practice*, Vol. 16, 2008, S. 1443-1456.
- [16] Z. Horvath und G. Hilbig, "Mathematical model for fuel-heated glass melting tanks", *Glastechnische Berichte*, Vol. 61, Nr. 10, 1988, S. 277-281.
- [17] G. Hilbig, "Three-dimensional analytical model for the straight-through flow in an all-electric glass melting furnace", *Glastechnische Berichte*, Vol. 63, Nr. 7, 1990, S. 185-192.
- [18] H.-J. Voss, "Mathematisches Modell zur Abschätzung des Energiehaushaltes von Glasschmelzwannenöfen", *Glastechnische Berichte*, Vol. 48, Nr. 9, 1975, S. 190-205.
- [19] G. Leyens, "Beitrag zur Berechnung zweidimensionaler Konvektionsströmungen in kontinuierlich betriebenen Glasschmelzwannen", *Glastechnische Berichte*, Vol. 47, Nr. 12, 1974, S. 251-270.
- [20] G. Leyens und R. Moreau, "Mathematische und meßtechnische Untersuchung verschiedener Glaswannen", *Glastechnische Berichte*, Vol. 51, Nr. 3, 1978, S. 43-47.
- [21] L. Mardorf und G. Woelk, "Berechnung des Verhaltens einer elektrisch beheizten Glasschmelzwanne mit einem mathematischen Modell", *Glastechnische Berichte*, Vol. 56, Nr. 4, 1983, S. 73-83.
-

- [22] G. Hilbig, "Zur Problematik der dreidimensionalen mathematischen Modellierung von Elektroglasschmelzwannen", *Glastechnische Berichte*, Vol. 57, Nr. 12, 1984, S. 301-306.
- [23] A. Urgan und R. Viskanta, "Three-dimensional numerical modeling of circulation and heat transfer in a glass melting tank: Part 1 - Mathematical formulation", *Glastechnische Berichte*, Vol. 60, Nr. 3, 1987, S. 71-77.
- [24] J. Studzinski, "Mathematical modelling of glass mass flow in a conventional glass tank furnace using isotope data", *Glastechnische Berichte*, Vol. 64, Nr. 7, 1991, S. 169-178.
- [25] J. Wang, B.S. Brewster, M.Q. McQuay und B.W. Webb, "Validation of an improved batch model in a coupled combustion space/melt tank/batch melting glass furnace simulation", *Glass Science Technology*, Vol. 73, Nr. 10, 2000, S. 299-307.
- [26] R. Beerkens, "Heutige Möglichkeiten zur Modellierung von Glasschmelzöfen: Voraussagen zur Qualität des Glasschmelzprozesses", *Glastechnische Berichte*, Vol. 71, Nr. 4, 1998, S. 35-47.
- [27] J. Bauer, U. Roger und P. Simons, "Validation of a mathematical glass tank model", *Glass Science Technology*, Vol. 72, Nr. 6, 1999, S. 171-181.
- [28] P. Astrid, *Reduction of Process Simulation Models: A proper orthogonal decomposition approach*. Dissertation an der Technischen Universität Eindhoven, 2004.
- [29] P.M. Nijse, *Modelling and control of a Glass Melting Furnace*. Masterarbeit an der Technischen Universität Eindhoven, 2004.
- [30] L. Huisman, *Control of Glass Melting Processes Based on Reduced CFD Models*. Dissertation an der Technischen Universität Eindhoven, 2005.
- [31] E. Muijsenberg, "Modeling of glass melting processes: How reliable & validated simulation tools can help to improve glass melting", *11. European Society of Glass Science (ESG) Konferenz in Maastricht*, Vortrag, 03.-06.06.2012.
- [32] F. Hegewald, H. Heelemann und P. Hemman, "Verfahren zur Messung und Regelung qualitätsbestimmender Parameter der Rauhschmelze von Glasschmelzwannen", Europäische Patentschrift der Software & Technologie Glas GmbH Cottbus, Patenterteilung 21.02.2007.
- [33] F. Hegewald, P. Hemman und H. Heelemann, "Verfahren zur Regelung der Lokalität der Flammenwärmeentbindung bei Wannenöfen", Offenlegungsschrift der Software & Technologie Glas GmbH Cottbus, DE10160828A1, Patenterteilung 08.08.2002.
- [34] P. Hemmann, A. Birle, T. Schulz und H. Heelemann, "Verfahren zum geregelten Betrieb eines regenerativ beheizten Industrieofens, Steuereinrichtung und Industrieofen", Offenlegungsschrift der Software & Technologie Glas GmbH Cottbus, DE102010041155A1, Patenterteilung 22.03.2012.



- [35] P. Hemmann, A. Birle, T. Schulz und H. Heelemann, "Verfahren zum geregelten Betrieb eines regenerativ beheizten Industrieofens, Steuereinrichtung und Industrieofen", auf dem Gebiet des Patentwesens (PCT) veröffentlichte internationale Anmeldung der Software & Technologie Glas GmbH, WO2012/038488A1, Patenterteilung 29.03.2012.
- [36] W. Stief, *Regelungstechnische Untersuchungen an einem Feeder*. Forschungsbericht der Fachhochschule Frankfurt/M, 2010.
- [37] M. Pohlscheid und C. Schmidt, "Regelungsvorrichtung für Brenner bzw. Verfahren zur Regelung von Brennern einer Glasschmelzwanne", Offenlegungsschrift der Siemens AG, DE19716106A1, München, Patenterteilung 22.10.1998.
- [38] B. Gough und D. Matovich, "Predictive-Adaptive Temperature Control of Molten Glass", *IEEE Industry Applications Society*, 1997, S. 51-55.
- [39] M. Huzmezan, B. Gough, S. Kovac, L. Le und G. Roberts, "A New Generation of Adaptive Model Based Predictive Controllers Applied in Batch Reactor Temperature Control", *IEEE Industry Applications Society*, 1999, S. 20-26.
- [40] H. Loch und D. Krause, *Mathematical Simulation in Glass Technology*. Mainz: Springer-Verlag Berlin Heidelberg 2002.
- [41] J. Chmelar, R. Bodi und E. Muijsenberg, "Supervisory advanced control of glass melters and forehearths by the GS expert system", *Glass Science Technology*, Vol. 73, Nr. 9, 2000, S. 276-284.
- [42] P. Schill, "Energy Savings and Glass Quality Improvement by Mathematical Modelling and Advanced Furnace Control", *Glassmann Europe 2007 Conference Proceedings*, S. 64-70.
- [43] R.R.J.J. Schobben, *Identification and CFD Model Based Control of Glass Furnace*. Masterarbeit an der Technischen Universität Eindhoven, 2009
- [44] Advanced Control Solutions Inc., "ACSI's Model-Based Controller Maximizes Performance in Glass Forehearths", Broschüre von Advanced Control Solutions Inc., zuletzt abgerufen unter <http://www.acsitoledo.com/FH%20Model%20Based%20Control.pdf> am 16.11.2015.
- [45] R. Finch, "Applications of Model Based Control in Float and Fiberglass", Broschüre von Advanced Control Solutions Inc., zuletzt abgerufen unter <http://www.acsitoledo.com/Glass%20Problems%20paper%202008.pdf> am 16.11.2015.
- [46] S. Nixon und M. Kornowa-Weichel, "Model Based Basics: An Overview of Model Based Control", Broschüre von Advanced Control Solutions Inc., zuletzt abgerufen unter <http://www.acsitoledo.com/MBC%20Overview.pdf> am 16.11.2015.

- [47] S. Nixon und M. Kornowa-Weichel, "Breaking from Tradition: Model Based Control vs. PID", Broschüre von Advanced Control Solutions Inc., zuletzt abgerufen unter <http://www.acsitoledo.com/PID%20vs.%20MBC.pdf> am 16.11.2015.
- [48] B. Fleischmann, "Regelungspraxis bei Feuerungsprozessen", Vortrag beim Seminar *Grundlagen der Feuerungstechnik zur Glasherstellung und Einfluss der Erdgasbeschaffenheit* der Hüttentechnischen Vereinigung der Deutschen Glasindustrie e. V. und des Gas- und Wärme-Instituts Essen e.V., Essen, 06.-07.05.2015.
- [49] P. Havel, "Overview of automatic control of glass furnaces", *Ceramics - Silikáty*, Vol. 50, Nr. 1, 2006, S. 51-55.
- [50] Firma SiliTec GmbH, "Viskosität als Prozess- und Regelgröße", *GLASS: Glasherstellung - Glasveredelung, Prozesse und Technologien*, 2. Edition, 2002, S. 42-51.
- [51] D. Schippan, *Untersuchung des reaktionstechnischen Verhaltens in Behälterglaswannen mit Tracerversuchen*. Dissertation an der Rheinisch-Westfälischen Technischen Hochschule Aachen, 2003, S. 92ff.
- [52] B. Fleischmann, "Einfluss von Gasbeschaffenheitsänderungen auf den Glasherstellungsprozess: Teil 1 - Aktuelle Situation", *Mitteilung der Hüttentechnischen Vereinigung der Deutschen Glasindustrie e. V.*, Nr. 2160, 2013, abrufbar unter <http://www.hvg-dgg.de/download/hvg-mitteilungen.html>.
- [53] A. Giese, "Auswirkungen von Gasbeschaffenheitsschwankungen auf industrielle Prozesse", Vortrag beim Seminar *Grundlagen der Feuerungstechnik zur Glasherstellung und Einfluss der Erdgasbeschaffenheit* der Hüttentechnischen Vereinigung der Deutschen Glasindustrie e. V. und des Gas- und Wärme-Instituts Essen e.V., Essen, 06.-07.05.2015.
- [54] "Künstliche Neuronale Netze", Forschungsschwerpunkte des Instituts der Regelungstechnik der Rheinisch-Westfälischen Technischen Hochschule Aachen, zuletzt abgerufen unter <http://www.irt.rwth-aachen.de/forschung/methoden/identifikation-dynamischer-systeme/knn/> am 16.11.2015.
- [55] E. Diaconescu, "The use of NARX Neural Networks to predict Chaotic Time Series", *WSEAS Transactions: World Scientific and Engineering Academy and Society*, Vol. 3, ISSN: 1991-8755, 2008, S. 182-191
- [56] K. Worden und R.J. Barthope, "Identification of Hysteretic System Using NARX Models: Part 1 - Evolutionary Identification", *Conference Proceedings of the Society for Experimental Mechanics*, Vol. 29, 2012, S.49-56.
- [57] K. Worden, R.J. Barthope und J.J. Hensman, "Identification of Hysteretic System Using NARX Models: Part 2 - A Bayesian Approach", *Conference Proceedings of the Society for Experimental Mechanics*, Vol. 29, 2012, S.57-65.

- 
- [58] X. Li und M. Lang, "Combination Prediction of Railway Freight Volume Based on Support Vector Machine and NARX Neural Network", *Proceedings of 3rd International Conference on Logistics, Informatics and Service Science*, Springer-Verlag 2015
- [59] N.S. Popov, D.J. Dozic, M. Stankovic, G.M. Krajoski und D. Stanisic, "Development of a Closed Loop FES System Based on NARX Radial Based Network", *IFMBE: International Federation of Medical and Biological Engineering Proceedings*, Vol. 50, S. 70-74.
- [60] X. Yang, H. Zhang und H. Zhou, "A Hybrid Methodology for Salinity Time Series Forecasting Based on Wavelet Transform and NARX Neural Networks", *Arabian Journal for Science and Engineering*, 2014, S. 6895-6905.
- [61] "Neuronale Netze in der Wasserwirtschaft", Abschlussbericht zum Verbundprojekt *Anwendung und Entwicklung neuronaler Verfahren zur Autonomen Prozesssteuerung (AENEAS)*, Förderkennzeichen: 01 IN 505 D, Zentrum für Neuroinformatik GmbH, Bochum, 1999
- [62] U. Ahle, "Künstliche Neuronale Netze für die Prognose des Fernwärmebedarfs", *Fuzzy Logik: Theorie und Praxis*, Springer-Verlag Berlin Heidelberg 1994, S. 292-299.
- [63] J. Biethahn, A. Hönerloh, J. Kuhl, M.-C. Leisewitz, V. Nissen und M. Tietze (Hrsg.), *Betriebswirtschaftliche Anwendungen des Soft Computing: Neuronale Netze, Fuzzy Systeme und Evolutionäre Algorithmen*. Braunschweig / Wiesbaden: Vieweg Verlag 1998, S. 79ff.
- [64] S. Zakharian, P. Ladewig-Riebler und S. Thoer, *Neuronale Netze für Ingenieure: Arbeits- und Übungsbuch für regelungstechnische Anwendungen*. Rüsselsheim: Vieweg Verlag Braunschweig / Wiesbaden 1998.
- [65] H.P.H. Anh und L.T. Loi, "Medical Image Classification and Symptoms Detection Using Fuzzy NARX Technique", *IFMBE: International Federation of Medical and Biological Engineering Proceedings*, Vol. 40, 2013, S. 335-342.
- [66] M.H. Beale, M.T. Hagan und H.B. Demuth, *Neural Network Toolbox: Getting Started Guide*. Benutzerhandbuch zur Software *MATLAB R2016a*, zuletzt abgerufen unter [http://de.mathworks.com/help/pdf\\_doc/nnet/nnet\\_gs.pdf](http://de.mathworks.com/help/pdf_doc/nnet/nnet_gs.pdf) (Login erforderlich) am 15.03.2016.
- [67] M.H. Beale, M.T. Hagan und H.B. Demut, *Neural Network Toolbox: User's Guide*. Benutzerhandbuch zur Software *MATALB R2016a*, zuletzt abgerufen unter [http://de.mathworks.com/help/pdf\\_doc/nnet/nnet\\_ug.pdf](http://de.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf) (Login erforderlich) am 15.03.2016.
- [68] A.A.A. Noureldin, *Process Identification and Control using Neural Networks and Fuzzy Logic*. Dissertation an der Brandenburgischen Technischen Universität Cottbus, 2003.
-

- [69] H. Yonaba, F. Anctil und V. Fortin, "Comparing Sigmoid Transfer Functions for Neural Network Multistep Ahead Streamflow Forecasting", *Journal of Hydrologic Engineering*, Vol. 15, Nr. 4, 2010, S. 275-283.
- [70] M. Nørgaard, O. Ravn, N.K. Poulsen und L.K. Kansen, *Neural Networks for Modelling and Control of Dynamic Systems*. Lyngby, Dänemark: Springer-Verlag London 2000.
- [71] T. Knohl, *Anwendung künstlicher neuronaler Netze zur nichtlinearen adaptiven Regelung*. Marl: VDI Verlag 2001, Fortschritt-Berichte VDI, Reihe 8, Nr. 885, S. 38ff.
- [72] H.-J. Zimmermann, *Neuro + Fuzzy: Technologien - Anwendungen*. Aachen: VDI-Verlag GmbH Düsseldorf 1995, S. 74ff.
- [73] R. Dittmar und B.-M. Pfeiffer, *Modellbasierte prädiktive Regelung: Eine Einführung für Ingenieure*. München: Oldenbourg Wissenschaftsverlag GmbH 2004.
- [74] J.M. Pina und P.U. Lima, "An Operation System for Industrial Processes: Application to a Glass Furnace", *Proceedings of the 10th Mediterranean Conference on Control and Automation - MED2002*, Portugal, 2002
- [75] P.H. Sørensen, M. Nørgaard, O. Ravn und N.K. Poulsen, "Implementation of neural network based non-linear predictive control", *Neurocomputing*, Vol. 28, 1999, S. 37-51.
- [76] A. Bemporad, M. Morari und N. Lawrence Ricker, *Model Predictive Control Toolbox: User's Guide*. Benutzerhandbuch zur Software MATLAB R2016a, zuletzt abgerufen unter [http://de.mathworks.com/help/pdf\\_doc/mpc/mpc\\_ug.pdf](http://de.mathworks.com/help/pdf_doc/mpc/mpc_ug.pdf) (Login erforderlich) am 15.03.2016.
- [77] A. Bollig, "Prädiktive Regelung mit Neuronalen Netzen", *at - Automatisierungstechnik*, Vol. 51, Nr. 02/2003, S. 69-77.
- [78] T. Apel, "Numerische Parameteridentifikation", Vorlesungsskript der Universität der Bundeswehr München, Stand 2012, zuletzt abgerufen unter <https://dokumente.unibw.de/pub/bscw.cgi/d6070941/parameter.pdf> am 03.03.2016.
- [79] B. von Harrach, "Einführung in die Optimierung", Vorlesungsskript der Universität Stuttgart, Stand 2015, zuletzt abgerufen unter <http://www.mathematik.uni-stuttgart.de/~harrach/lehre/Optimierung.pdf> am 03.03.2016.
- [80] H.P. Gavin, "The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems", Veröffentlichung des Departments of Civil and Environmental Engineering der Duke University, 29.09.2015, zuletzt abgerufen unter <http://people.duke.edu/~hpgavin/ce281/lm.pdf> am 03.03.2016.
- [81] M.K. Transtrum und J.P. Sethna, "Improvements to the Levenberg-Marquardt algorithm for nonlinear least-squares minimization", *Journal of Computational Physics*, 30.01.2012, Cornell University Library, zuletzt abgerufen unter <http://arxiv.org/pdf/1201.5885v1.pdf> am 03.03.2016.

- [82] J.M. Maciejowski, *Predictive Control with Constraints*. Harlow, England: Prentice Hall 2002.